μSystems Research Group
School of Engineering

Memristor-based design solutions for mitigating parametric variations in IoT applications

Thanasin Bunnam

Technical Report Series
NCL-EEE-MICRO-TR-2021-220

June 2021

Contact: thanasin.b@gmail.com

Supported by EPSRC grants EP/N023641/1 & EP/L025507/1

NCL-EEE-MICRO-TR-2021-220

μSystems Research Group
School of Engineering
Merz Court
Newcastle University
Newcastle upon Tyne, NE1 7RU, UK

http://async.org.uk/

Abstract

Rapid advancement of the internet of things (IoT) is predicated by two important factors of the electronic technology, namely device size and energy-efficiency. With smaller size comes the problem of process, voltage and temperature (PVT) variations of delays which are the key operational parameters of devices. Parametric variability is also an obstacle on the way to allowing devices to work in systems with unpredictable power sources, such as those powered by energy-harvesters. Designers tackle these problems holistically by developing new techniques such as asynchronous logic, where mechanisms such as matching delays are widely used to adapt to delay variations. To mitigate energy efficiency and power interruption issues the matching delays need to be ideally retained in a non-volatile storage. Meanwhile, a resistive memory called memristor becomes a promising component for power-restricted applications owing to its inherent non-volatility. While providing non-volatility, the use of memristor in delay matching incurs some power overheads. This creates the first challenge on the way of introducing memristors into IoT devices for the delay matching.

Another important factor affecting the use of memristors in IoT devices is the dependence of the memristor value on temperature. For example, a memristance decoder used in the memristor-based components must be able to correct the read data without incurring significant overheads on the overall system. This creates the second challenge for overcoming the temperature effect in memristance decoding process.

In this research, we propose methods for improving PVT tolerance and energy characteristics of IoT devices from the perspective of above two main challenges: (i) utilising memristor to enhance the energy efficiency of the delay element (DE), and (ii) improving the temperature awareness and energy robustness of the memristance decoder.

For memristor-based delay element (MemDE), we applied a memristor between two inverters to vary the path resistance, which determines the RC delay. This allows power saving due to the low number of switching components and the absence of external delay storage. We also investigate a solution for avoiding the unintended tuning (UT) and a timing model to estimate the proper pulse width for memristance tuning. The simulation results based on UMC 180nm technology and VTEAM model show the MemDE can provide the delay between 0.55ns and 1.44ns which is compatible to the 4-bit multiplexer-based delay element (MuxDE) in the same technology while consuming thirteen times less power. The key contribution within (i) is the development of low-power MemDE to mitigate the timing mismatch caused by PVT variations.

To estimate the temperature effect on memristance, we develop an empirical temperature model which fits both titanium dioxide and silver chalcogenide memristors. The temperature experiments are conducted using the latter device, and the results confirm the validity of the proposed model with the accuracy R-squared $>$88%. The memristance decoder is designed to deliver two key advantages. Firstly, the temperature model is integrated into the VTEAM model to enable the temperature compensation. Secondly, it supports resolution scalability to match the energy budget. The simulation results of the 2-bit decoder based on UMC 65nm technology show the energy can be varied between 49fJ and 98fJ. This is the second major contribution to address the challenge (ii).

This thesis gives future research directions into an in-depth study of the memristive electronics as a variation-robust energy-efficient design paradigm and its impact on developing future IoT applications.

Contents

List of Figures

List of Tables

List of Acronyms

BD - Bundled-Data

BEOL - Back End Of Line

CBRAM -Conductive Bridge Random-Access Memory

CMOS - Complementary metal-oxide-semiconductor

CSDE - Current-Starved Delay Element

DCCS - Digitally Controlled Current Source

DE - Delay Element

DR - Dual-Rail

DRAM - Dynamic Random-Access Memory

EDR - Error Detection Register

FEOL - Front End Of Line

GMET - General-purpose MEmristance Tuning

HCI - Hot Carrier Injection

I/O - Input/Output

IoT - Internet of Things

LER - Line Edge Roughness

MemDE - Memristor-based Delay Element

MOSFET - Metal Oxide Semiconductor Field Effect Transistor

MSR - MetaStability Resolver

MuxDE - Multiplexer-based Delay Element

NBTI - Negative Bias Temperature Instability

NMOS - N-channel Metal Oxide Semiconductor

NRZ - Non-Return to Zero

PBTI - Positive Bias Temperature Instability

PDT - Post-fabrication Delay Tuning

PMOS - P-channel Metal Oxide Semiconductor

PVT - Process, Voltage and Temperature

RC - Resistance and Capacitance

RDF - Random Dopant Fluctuation

RTZ - Return To Zero

SET - Single-Event Transient

SEU - Single-Event Upset

SRAM - Static Random-Access Memory

TDDB - Time-Dependent Dielectric Breakdown

TID - Total Ionising Dose

TS - Timing Speculation

UT - Unintended Tuning

VTEAM - Voltage ThrEshold Adaptive Memristor

Acknowledgments

I would like to express my gratitude to Alex Yakovlev, my supervisor, for his wisdom and guidance throughout my postgraduate studies. He educated me to become a better researcher. I would also like to thank my second supervisor, Danil Sokolov, for helping me in improving the quality of my publications, and providing invaluable guidance during my research. I am grateful to Arthit Thongtak, who supervised me during my Masters studies; he introduced me to the world of asynchronous circuits.

I also extend my thanks to Ahmed Soltan, Oleg Maevsky and Patrick Degenaar, who acted as my unofficial supervisors. Ahmed Soltan shared his huge experience in the field of analog circuit design. The methods proposed in Chapters 4 and 5 were developed in close collaboration with him. Oleg Maevsky inspired me in the design of the memristance decoder proposed in Chapter 5. Patrick Degenaar gave me priceless advice about the experiments in Chapter 4 and allowed me to bring his lab equipment to my flat during lockdown. I would like to thank Adrian Wheeldon, Jedsada Phengsuwan, Titayapa Meenapinunt, Jirat Mankasem and Stephen Robson for helping me to move the lab equipment. I would also like to thank Eric Johnson from Knowm Inc. for providing me with useful information regarding silver-chalcogenide memristors.

I am thankful to my wife, Oranich Bunnam, for all of her love, support and patience throughout my PhD. She was there to help me at difficult times, and to share in the good times. I am grateful to my family for the concern and support all the time. Special thanks go to my brother, Ekalak Bunnam, for taking care of my dog, Keaw, while I've been pursuing my degree. My thanks also to all my friends, especially to Adrian Wheeldon, Serhii Mileiko and Shengqi Yu, who made my PhD journey even more fun by involving me into various sports and social activities.

This research was supported by EPSRC, grants STRATA (EP/N023641/1) and A4A (EP/L025507/1). My PhD scholarship was sponsored by the Royal Thai Government.

Chapter 1

Introduction

1.1 Challenges of IoT applications

The number of IoT devices is growing rapidly due to the wide range of applications and fields concerned, such as agriculture, healthcare and disaster warning. Next-generation IoT devices are expected to incorporate improvements in the following areas:

- Higher intelligence – machine learning techniques will be integrated so that decisions are made by the devices themselves, giving shorter response times and lower power loss in data transmission [1, 2].
- More compact size – where advanced features such as machine learning can be integrated with a small form factor [2]. The convolutional neural networks proposed by Yakopcic *et al.* [3] and Ni *et al.* [4] occupy about 1mm^2 chip areas.
- Lower power use – devices are typically equipped with batteries or energy harvesters; therefore, power reductions can improve system reliability [1, 2]. Blaauw *et al.* reveal the millimetre-size sensing devices supplied by 1mm^2 thin-film lithium batteries must consume average power in nano-Watts to be able to live for a year [5].
- Low maintenance requirements – maintenance capability and cost become issues when many millions of devices operate worldwide. This is more serious in pandemic situations when mobility is restricted due to lockdown measures.

These improvements rely on the two key factors of device size and energy efficiency, which can be achieved by *technology scaling*.

As shown in Fig. 1.1, when a technology node is reduced, the following parametric variations become more critical:

- Process variation – imperfections in the manufacturing process cause deviations in transistor dimensions. This contributes to variability in the threshold voltage and output driving strength. Pelgrom *et al.* proposed the standard deviation of the transistor's threshold is inversely proportional to the squared root of the gate area [6]. This relation is confirmed by the model proposed by Ye *et al.*, which estimates the threshold's standard deviation of 120mV when the technology is scaled down to 12nm [7].

- Voltage variation – unreliable power sources such as energy harvesters lead to fluctuations in supply voltages. Zhang *et al.* reported the output voltage of the on-chip switching capacitor DC-DC converter in UMC 90nm technology swings between 0.6V$-$1V [8]. This causes a variation in the transistor's gate-source voltage which has a quadratic relationship to its output current (drain-source current) [9].

- Temperature variation – changes in temperature mainly affect the transistor's threshold voltage and carrier mobility. The temperature dependency of the threshold voltage is negative and near-linear, while the carrier mobility is typically proportional to $T^{-3/2}$ [9], where T stands for temperature in Kelvin. The temperature variation leads to uncertain output current.

- Aging – the threshold voltage of the transistor increases over time due to the effects of negative/positive bias temperature instability (N/PBTI) and hot carrier injection (HCI). The increased threshold degrades the output current resulting in slower speed. Ghosh and Roy reveal the mean inverter gate delay in PTM 32nm technology rises by $\approx$9% within three years [10]. Additionally, the wire shape can be distorted by the electromigration effect increasing the wire resistance [11].

In summary, all of the effects mentioned above contribute to uncertainty associated with the transistor's output current and consequently severely affect path delays. Delay mismatches cause timing violations when data propagate to the destination register later

than the clock signal [12]. Therefore, to decrease feature size successfully, *timing violations* need to be mitigated.

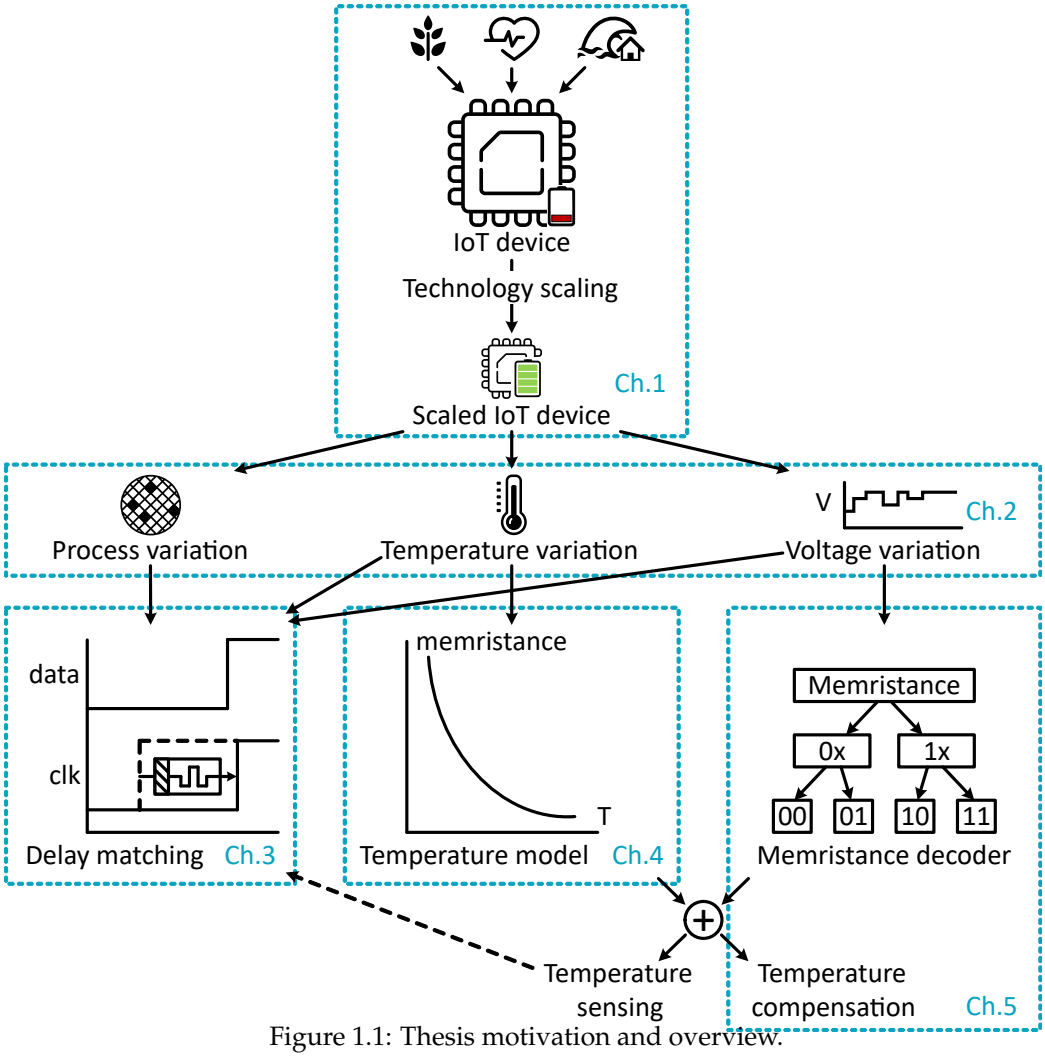


Figure 1.1: Thesis motivation and overview.

1.2 The potential use of memristors to mitigate timing violations

Typically, problems of timing violation are solved by adjusting the clock signal to satisfy the setup and hold times of the receivers [13, 14] or detecting the timing violation and recovering the system accordingly [15, 16]. Although the above methods can solve the problems, the system frequency is degraded to cover the worst-case scenario where the timing violation occurs at the longest path. Alternatively, asynchronous logic has been proposed to tackle timing issues by using adaptive delay lines instead of clock

signals [17]. This technique can solve the problem and leverage system performance to the average case due to its data-driven nature [18, 19]. For example, the latency of an asynchronous ripple-carry adder is considered in term of average value because the latency can vary between the shortest value (best) when the carry is not generated and the longest value (worst) in case the carry has to propagate through the entire carry chain [20]. On the contrary, the synchronous adder's latency must be selected from the worst-case value even the carry does not exist due to the lack of completion detection. In [17], the margin penalty of the asynchronous system increases only 13% at a 40% rate of timing violations. In comparison, its synchronous counterpart's margin penalty increases as high as 30%, with just a 10% rising in the rate of timing violations. Although asynchronous logic offers higher performance and more robustness to parametric variability, further configurable delay circuitry is required. The subsequent higher power consumption then leads to greater risks of configuration loss due to power interruption [17]. This extra power is not spent only for switching the delay circuitry, but it is also used for transferring the delay configuration to and from the memory device. Dally reported the energy used for data movement is 3 order of magnitude higher than that of the computing task [21]. Therefore, retaining the delay configuration in *non-volatile storage* without the need for *data movement* would be an ideal solution.

Recently, a resistive memory device called the memristor has been invented [22, 23]. Its most interesting feature is that its internal resistance, called memristance, can be tuned and retained owing to its inherent non-volatility. In comparison to NAND flash, which is a current non-volatile memory technology, the memristor has a similar data retention of 10 years [24–26]. In addition, Stathopoulos *et al.* [27] showed the Al_xO_y/TiO_2 device is stable at high temperature (85°C) for 8 hours. The high endurance of 10^{12} cycles, which is 7-8 orders magnitude higher than NAND flash, was reported by Yang *et al.* [24] and Abunahla *et al.* [28]. Considering the memristor's performance, it can be read and written within 10ns which is 4 orders magnitude faster than NAND flash and close to the speed of DRAM [24, 26, 28]. Although the operating speed of SRAM, which is in sub-nanoseconds, is faster than that of the memristor, it requires up to 35x larger area [24,26,28]. Lastly, the memristor can be fabricated using the existing CMOS process as demonstrated by Maheshwari *et al.* [29]; it was formed between two metal layers on top of the transistor. Due to the aforementioned features, memristor is considered to be

a promising device for power-restricted applications [30].

Regarding RC delay, digital circuits' propagation times can be controlled by adjusting either path resistance or path capacitance. To increase the delay, adding path resistance provides a better power saving because the current can be reduced. Adding path capacitance, however, requires more charges to fill up the capacitor resulting in higher power consumption. Because delay is proportional to resistance and the memristor's resistance is non-volatile, this device creates an opportunity to develop a *low-power delay matching* solution to mitigate the timing violation problem in IoT devices.

1.3 Requirements for mitigation of temperature effects in memristor-based IoT devices

The memristor is also affected by the following variations:

- Process variation – imperfections in memristor geometry cause deviations in memristance range, threshold and switching time [31]. Hu *et al.* reported the OFF and ON memristances of titanium dioxide memristor vary between -5.5% and 4.8% when the deviation of the device's thickness is 2% [32]. The experimental data from Li *et al.* show an inverse non-linear relationship between active areas and set voltages [33]. Therefore, the uncertainty in the fabrication process causes the threshold voltage variability and consequently affects the switching time as expressed by Kvatinsky *et al.* [34].

- Voltage variation – the memristance can be programmed by applying a voltage above the memristor's threshold for an amount of time; thus, the tuning accuracy decreases when at least one of both parameters deviates [34]. Importantly, the phase of the material, which dominates the memristance, can be affected by Joule heating resulting from the over-supply voltage [25]. When a voltage below the threshold is applied for a memristance reading, this variation may lift the signal amplitude and cause unexpected programming if this amplitude exceeds the threshold.

- Temperature variation – changes in temperature affect the value of memristance in both volatile and non-volatile fashions. The volatility comes from the temperature dependency of the material's resistivity [31, 35–40]. The temperature can also

change the phase of the material, and this change is non-volatile [25]. Although both effects shift the memristance and consequently cause reading errors, the non-volatile effect is less significant due to the fact that a sufficiently high temperature is required to start the phase transition. For example, chalcogenide (Ge_2Se_3) material needs about 350°C to start the crystallisation, which reduces the resistivity [41]. The memristor switching time is also reported as temperature-dependent. It decreases non-linearly when the temperature rises as described by Abunahla *et al.* [42].

Among these variations, the temperature effect is the worst problem because it has a high impact on memristance, as depicted in Fig. 1.1. Hence, error still occurs even when a large memristance margin is adopted and a high-precision programming technique is implemented. This creates a requirement of temperature compensation for the memristance decoding process, which is essential in memristor applications such as crossbar dot-product engines [1, 43, 44], neuromorphic computing [45–47], biosensors [48–50], and multi-bit memories [27, 51–54]. Alternatively, temperature sensitivity can be utilised to build a temperature sensor which can cooperate with the clock generator to adjust clock frequency in order to prevent timing violations. Both methods create a challenge to model the *temperature sensitivity* of the memristor so as to provide a solution to mitigate the temperature effect.

The memristance decoder is an essential building block in reading the memristor's value. It is composed of a comparator which is used to compare the memristance against reference values. To decode the memristance, the comparison process is iterated using the reference values to satisfy the desired output resolution. Therefore, to achieve high resolution, energy consumption is increased. This causes a reliability issue in modern IoT devices because they are typically equipped with an unpredictable power source such as an energy harvester. To operate under these conditions, the design of a *memristance decoder* that can scale its resolution based on the available energy budget is challenging. Coupling this decoder with the temperature model can thus create an energy-adaptive solution for temperature compensation and sensing, which are necessary in developing memristor-based IoT devices.

1.4 Summary

The future development of IoT devices will be directed towards reducing their size and improving energy efficiency. Technology scaling can achieve both goals holistically, but risks failure due to the effects of parametric variations. In this research, the problems of such variations are approached from two perspectives:

(i) We propose methods for utilising memristor to improve the energy efficiency of delay matching in CMOS circuits;

(ii) We propose methods for improving the temperature awareness and energy robustness of the memristance decoder.

1.5 Contributions

The main contributions of this thesis are as follows:

- **Pulse-controlled memristor-based delay element design**
  A memristor-based delay element is implemented by placing the memristor between two inverters to vary the path resistance, which affects the delay. We further investigate a solution to the unintended tuning problem that occurs when the signal level is higher than one of the memristor's thresholds. To determine the appropriate pulse width for memristance tuning, an excitation time model based on a voltage divider is realised.

  The main advantage of this delay element is energy efficiency due to the smaller number of transistors required and the absence of the current path in normal mode, where the input propagates to the output with some delay. Furthermore, the proposed delay element does not need to be re-initialised every time the system starts, since the delay can be retained in the memristor. Thus, the delay element can mitigate timing violations while saving on overall energy consumption and reducing the start-up time of IoT devices. This work relates to perspective (i).

- **Temperature model of the memristors**
  An empirical-based temperature sensitivity of the titanium dioxide memristor is modelled and integrated into the VTEAM model. The model describes the

relationship between OFF memristance and temperature in an inverse exponential equation. This model is further applied to the silver-chalcogenide memristor, which is the only off-the-shelf memristor currently available. The experimental results from measurements of OFF memristance during a temperature sweep confirm the model's accuracy, with a value of R-squared $> 88\%$.

The model presented is expected to cover the memristors that are build from metal-oxide and chalcogenide materials. It will enable improved temperature tolerance in the design of memristive circuits and temperature sensing applications. This work is combined with the proposed memristance decoder to address perspective (ii).

- **Design of thermally-aware memristance decoder**

  A resolution-scalable memristance decoder circuit is proposed to support memristance decoding based on the energy budget. The comparator is the main building block, which is designed based on the current-mode circuit to achieve high performance at low power. The temperature model is applied in a simulation to demonstrate the decoding error due to the temperature effect and data correction. Furthermore, a configurable capacitor array is implemented to mitigate the offset from process variation. Besides, the decoder supports both synchronous and asynchronous schemes and does not contain any resistors which are susceptible to parametric variation.

  Ultimately, the proposed decoder is suitable for memristor-based IoT devices where power and parametric variation affect their operation. This work deals with perspective (ii).

1.6 Thesis layout

This thesis is organised as follows:

Chapter 1 – Introduction. This chapter briefly discusses the background and motivation for the thesis and summarises its contributions.

Chapter 2 – Background. An overview is given of parametric variations, methods of asynchronous circuit design and the requirements of delay elements. The major challenges of delay element designs to address timing violations are described. We summarise the properties of the memristor, which is used to design a delay element in

this study. The opportunity to model the temperature effect is discussed. Memristance decoders and their potential to compensate for the effects of temperature on memristance are investigated.

Chapter 3 – Memristor-based delay elements. In this chapter, a method is developed for memristor-based delay element design for asynchronous bundled data circuits. A method of unintended tuning avoidance and an excitation time model to estimate the proper pulse width for memristance tuning are also developed.

Chapter 4 – Modelling temperature effect on the memristance. This chapter proposes an empirical model to estimate changes in memristance with respect to temperature, and it includes a theoretical analysis supported by experimental results.

Chapter 5 – Thermally-aware memristance decoder. A method of memristance decoder design with resolution scalability is developed. The memristor model from chapter 4 is applied to demonstrate the temperature effect and its compensation.

Chapter 6 – Conclusions. The contributions of the study discussed in this thesis are summarised, and future research areas for the development of memristor-based design solutions for mitigating parametric variations in IoT applications are suggested.

Chapter 2

Background

This chapter reviews current achievements in the field subject to investigation. Section 2.1 describes the sources and effects of parametric variations. It explains the techniques used to improve robustness against such variations, beginning with an explanation of timing violation in synchronous circuits and existing solutions, including the use of a delay element. Then, asynchronous circuits and their handshaking protocols are considered. The role of the delay element for bundled-data protocols is explained. Finally, the performance of the techniques mentioned are discussed and the opportunity provided by a bundled-data protocol, which requires a delay element, is highlighted.

In section 2.2, existing delay elements are reviewed. The advantages of memristor-based delay elements compared to CMOS ones are discussed. The improvements needed for a memristor-based delay element are highlighted.

Section 2.3 describes the memristive devices which are built based on titanium dioxide and chalcogenide materials. The memristor model called VTEAM and its parameter sets are explained. The challenges in temperature modelling and tuning pulse width are discussed. Finally, the memristor's metastability is reviewed.

In section 2.4, the need of comparator in memristance decoding process is explained. The design, operation and opportunity of the latch comparator is considered as well as its offset and metastability issues.

Section 2.5 summarises the impact of parametric variations on IoT devices which are based on digital and memristive systems. The opportunities to develop a memristor-

based delay element, modelling of the temperature effect on memristance, and the design of a memristance decoder to mitigate such variations are highlighted.

2.1 Parametric variations-tolerant design techniques

2.1.1 Parametric variations in CMOS circuits

Parametric variations can be classified into two categories: process variations (static) and dynamic variations. Process variation tends to increase when feature size is reduced, due to limitations in manufacturing processes; for example, subwavelength lithography, random dopant fluctuations (RDFs), and line edge roughness (LER). In the optical lithography process, the wavelength of light used to create a pattern is only slowly scaled compared to feature size, as illustrated in Fig. 2.1. The wavelengths used started to become smaller in 1994 when feature size had reached 0.35μm and this gap has steadily increased since that time. This issue causes loss of accuracy in layout printing and consequently imperfections in transistor dimensions, oxide thickness and threshold voltage. RDF causes variability in dopant atom concentration. In the older technologies, the large doping area could accept a high number of doping atoms. Therefore, small fluctuations in the number of these atoms did not affect their concentration. However, as transistor size has been scaled down, the number of doping atoms has declined, as shown in Fig. 2.2. Consequently, a small variation in dopant atoms has a bigger impact on the concentration, resulting in threshold voltage variation [55, 56]. LER arises from imperfections in the etching process. The imprecise etching of the poly layer results in different channel widths and lengths of the transistors that are supposed to be identical. This causes difference in transistor driving strength. Overall, technology scaling increases process variation and consequently affects the threshold and driving strength of the transistors.

Dynamic variation includes the effect of environmental variations and ageing. Variability in environmental conditions can be further classified as voltage, temperature, humidity and radiation variations. In large technology nodes, the effect of voltage variations is not significant since the supply voltage is relatively high compared to the transistor's threshold. However, this high voltage causes high power consumption, which is not suitable for IoT applications which are operated using limited and unstable

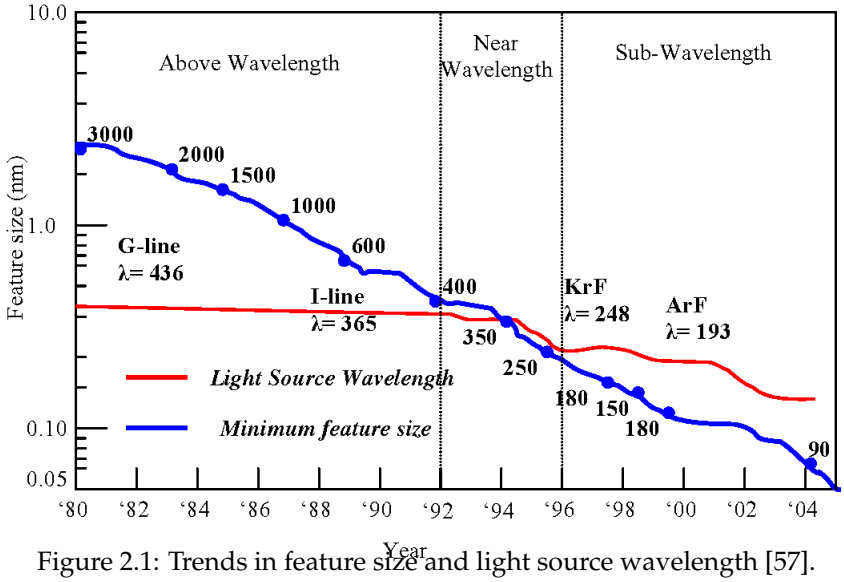


Figure 2.1: Trends in feature size and light source wavelength [57].

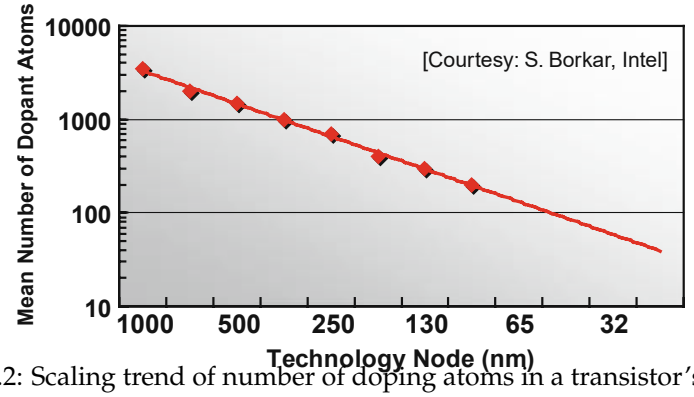


Figure 2.2: Scaling trend of number of doping atoms in a transistor's channel.

power sources. Scaling the technology down can reduce the operating voltage and power dissipation, but the gap between the supply voltage and transistor threshold is also reduced, and thus variations in supply voltage may severely affect circuit performance, as shown in Fig. 2.3. It can be seen that the standard deviation of the path delay is increased when the supply voltage (V_{dd}) falls below the nominal value and hence the number of unqualified chips (striped area) is expanded.

Temperature variation involves not only ambient temperature but also spatial heat dissipation, as shown in Fig. 2.4. The areas that undergo higher switching activity dissipate more power than others, and this power is transformed into the heat. Differences in temperature between areas alter the transistor thresholds at different rates. The thresholds of the transistors in hotter areas are reduced, and hence their switching speeds are faster than those in colder areas. Therefore, temperature variation causes

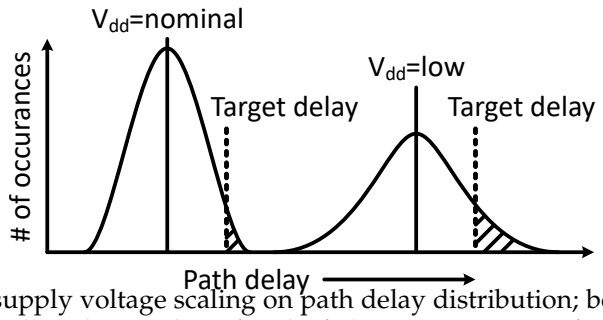


Figure 2.3: Impact of supply voltage scaling on path delay distribution; both mean and sigma of delay distribution as well as the number of paths failing to meet target frequency increase [10].

timing mismatches in every part of the circuit. In addition, although the reduction in threshold voltage at high temperature accelerates performance, it increases the leakage current which contributes to heating and power loss, and lowers the stability of IoT devices.

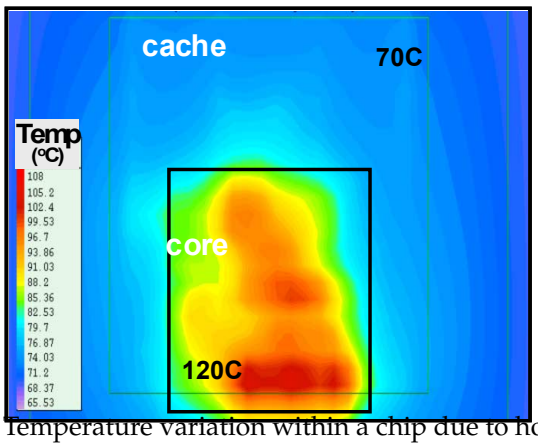


Figure 2.4: Temperature variation within a chip due to hot-spot [58].

In integrated circuits, moisture causes corrosion due to electrochemical migration between wires, and performance degradation regarding the increased dielectric constant [59]. However, Stellari *et al.* show the interconnect deterioration decreases the overall performance of ring oscillators and voltage-controlled oscillators by only less than 3% [60]. This is because the performance degradation is dominated by the front end of line (FEOL) devices, i.e. transistors, which are protected from moisture by the interconnect layers. Although the humidity impact on performance is insignificant, the impact on reliability is still concerned, especially in designing implantable devices which operate in humid environments. Eder *et al.* propose temperature and humidity sensors with a small footprint ($0.2mm^2$) for monitoring both parameters in realtime [61].

The radiation can impact CMOS devices in the short and long terms. Total ionising

dose (TID) contributes charge trapping in the oxide layer and silicon-oxide interface which can shift the transistor's threshold voltage in long term. Both oxide- and interface-trapped charges are positive in the PMOS transistor therefore its threshold is decreased. In the case of NMOS transistor, only interface-trapped charge is negative leading to a slower increase in threshold voltage. Borghello observes, based on 65nm technology, the PMOS transistor fails to conduct current once the irradiation reaches 1 Grad(SiO_2) at T=298K. In contrast, the current driving strength of the NMOS transistor reduces by 60% at the same irradiation value [62]. Single event effect (SEE) [63] can cause a signal switching which results in the wrong data collection if the register is triggered during this accidental switching (single-event transient, SET). It can also flip the logic inside the register leading to the data error (single-event upset, SEU).

Ageing is an effect where circuit performance degrades over time. It consists of a number of phenomena such as negative bias temperature instability (NBTI), positive bias temperature instability (PBTI), hot carrier injection (HCI), time-dependent dielectric breakdown (TDDB) and electromigration. Ageing becomes a serious problem when gate oxide thickness is scaled to below 1.5nm [64, 65].

NBTI occurs when applying negative gate-source voltage to a PMOS transistor. This votlage creates traps at the substrate-dielectric interface which capture the electrons from the channel, resulting in a rise in threshold voltage and timing mismatch. Removing the gate-source voltage can release electrons from these traps, but some still remain. Therefore, the PMOS switching time increases over time. The mechanism of PBTI is the same as for NBTI except that it happens in NMOS transistors biased by a positive gate-source voltage. Both mechanisms are magnified at high temperature [66]. Although the impact of PBTI is negligible compared to NBTI, it became a serious issue once the Hi-K/Metal Gate transistor was introduced [67].

HCI is similar to N/PBTI but in a different direction. It occurs when applying a horizontal electric field between drain and source regions. This electric field accelerates electrons into the channel and those that have enough energy (hot carriers) will enter dielectric and substrate permanently. Another difference between HCI and N/PBTI is that the latter occurs when the transistors are ON ($|Vgs| > 0$), while HCI happens when a transistor switches from low to high. Therefore, the HCI effect increases with operating frequency [10]. In addition, this effect has an exponential relationship to

supply voltage [68] while temperature does not magnify it crucially [69]. HCI usually arises in NMOSs because the major carrier is the electron. It results in performance degradation due to the elevated transistor threshold. Schlunder *et al.* reveal N/PBTI dominates the frequency degradation of ring oscillators while the HCI effect is clearly observed after 10 years of operation [70].

TDDB or oxide breakdown [71] is a corollary of N/PBTI and HCI. The increasing amount of trapped charges inside the dielectric over time can create a conductive path between the gate and channel of the transistor which results in functional failure. The risk of this effect increases when the technology is scaled down because the dielectric layer is thinner.

Electromigration is an effect that distorts the shape of wire due to the transfer of momentum between conducting electrons and diffusing metal atoms [11]. A wire that contains a high current density is vulnerable to this effect. When the wire is mis-shaped, its resistance increases resulting in longer path delay.

In summary, technology scaling is beneficial to IoT applications because it can reduce device size and power consumption. However, this comes at the cost of parametric variations which lead to timing mismatches in digital circuits. In particular, the timing is changed dynamically due to environmental variation and ageing. Therefore, an online delay matching technique is essential in tackling these variations. This technique also needs to be energy-efficient so as to suit IoT applications.

2.1.2 Techniques for synchronous circuits

In synchronous circuits, parameter variations alter the path resistances of the clock signal so that it cannot satisfy the setup and hold times of the receivers. For example, considering the synchronous pipeline shown in Fig. 2.5a, the combinational circuit processes the input from register A (RegA) and the output is collected by register B (RegB). Regarding the timing diagram in Fig. 2.5b, ideally, the clock period is assigned to satisfy the register's setup time, which is the time that the input signal must be stable before the clock transition. In practical, however, the propagation time of the combinational block may delay due to the effects of these variations. As a result, the setup time is unsatisfied and the data correctness cannot be guaranteed.

Traditionally, a safety margin is added to the clock period to ensure the circuit can still

operate during the presence of variations. For TSMC 0.18μm technology, the clock timing must be delayed by 27% of the nominal value to cover the worst-case performance (slow-slow process corner) due to the process variation. Besides, the effects of low voltage and high temperature drop the transistor's performance by 13%. Accumulating the above effects results in a larger margin which is 62% of the nominal value [9]. In designing a 64-bit processor based on 0.13μm technology, Das *et al.* show a 150ps margin (43%) is added to the pulse width of 350ps to prevent the variation-induced timing violation conservatively [15].

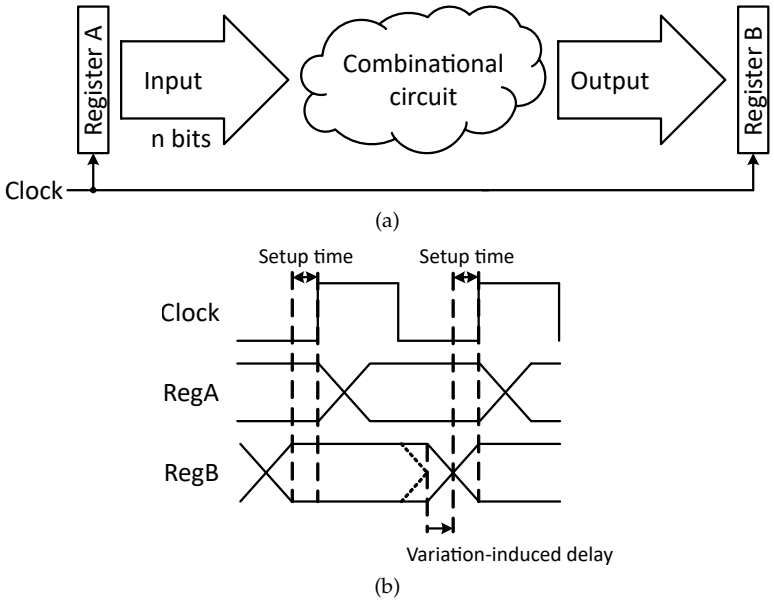


Figure 2.5: Synchronous pipeline: (a) pipeline structure; (b) timing diagram. The input of the B register (RegB) is stable before the first clock transition by the setup time; therefore, RegB can store the data correctly. In contrast, if this input delays due to the parametric variations (the second clock transition), the setup time may not satisfy, and RegB may collect the wrong data.

There are several solutions to this timing mismatch. Post-fabrication delay tuning (PDT), where the delays in each chip are set individually before shipping, has been proposed [13, 14]. However, this requires an extra step in the manufacturing process to assess the timing of the chip one-by-one. This increases production costs. Furthermore, because this delay is tailored for process variation only, a large delay margin is still required to increase the fault coverage with regard to temperature and voltage variations.

Timing speculation (TS) is another solution to the variation problem. In this method, the clock speed is elevated to a level such that timing violation occurs only occasionally. Some special components, such as Razor flip-flops (RazorFFs) [15, 16] are employed to

detect the error and recover the state of the system accordingly. Although this method offers better-than-worst-case performance, runtime variation can increase the error rate, so that the system spends most of the time in error recovery.

In another possibility, timing speculation and online delay tuning techniques are combined [72]. This is done to vary the path timing using delay elements when RazorFFs detect the error. Therefore, the timing of the critical paths can be adjusted to maintain an acceptable error rate. However, this method is not suitable for power-restricted applications because extra power is needed in retaining the delay configurations.

From the aforementioned methods, we can conclude that delay elements play a crucial role in mitigating the effect of parametric variations. However, operating the delay elements and maintaining their configurations requires extra power. This power overhead should be spent on improving the performance, and not for delay.

2.1.3 Asynchronous circuits

The idea of asynchronous circuitry arises because of the timing issues in clock-based circuits due to parametric variations. Removing the clock signal gains the following advantages [18, 19]:

- Low power consumption – the circuits are activated only when data arrives. Therefore, asynchronous circuits are known as data-driven circuits.
- Average-case performance – every sub-circuit can perform at its best speed and there is no need to wait for unused paths.
- Parametric variations robustness – the handshaking protocols are insensitive to timing mismatch.
- No clock distribution and clock skew problems.

A handshaking protocol is essential for data validation when the clock signal is absent. Two types of protocol are well-known in the literature: bundled-data (BD) and dual-rail (DR).

Bundled-data protocol

As depicted in Fig. 2.6a, the BD system is similar to the synchronous one except that synchronisation is accomplished using request (*Req*) and acknowledge (*Ack*) signals instead of clock signal. A BD circuit can be implemented based on a 4-phase or 2-phase data transfer scheme. The operating sequence of 4-phase BD can be described as in the timing diagram in Fig. 2.6b. When RegB is ready ($Ack = 0$), RegA will send the data to the combinational circuit and send a *Req* signal ($Req.A$) to RegB to request the collection of the result. Once RegB receives the *Req* signal ($Req.B$), it stores the data and sends an *Ack* signal to RegA to confirm that the data is collected. Then, RegA will reset the *Req* signal and consequently RegB will acknowledge by resetting the *Ack* signal. At this point, the pipeline is ready to process the new data. In the literature, this scheme is also called return-to-zero (RTZ) signalling because both *Req* and *Ack* must be returned to 0 before transfering new data. Although 4-phase is a simple scheme, its throughput is low because there is no processing task while resetting the *Req* and *Ack* signals.

To increase throughput, a 2-phase data transfer scheme has been proposed. Unlike 4-phase, data transfer is now based on signal transition instead of logic level, as shown in Fig. 2.6c. When *Ack* toggles, RegA will release the data into the combinational circuit and switch the *Req* signal. Once RegB detects a *Req* transition, it toggles *Ack* signal to inform RegA that it is ready for new data and the whole process is repeated. It can be seen that the throughput of this scheme is higher than the previous one as the pipeline is never in an idle state. This scheme is also called non-return-to-zero (NRZ) in the literature. An implementation of 2-phase BD can be found elsewhere [17].

Delay elements are crucial for BD protocols because *Req* must be delayed by at least the propagation time of the datapath in order to ensure that RegB receives valid data. In particular, the delays for both rising and falling transitions should be the same in a 2-phase scheme because data is sent to the combination circuit in every transition.

The issue in delay assignment is similar to that in synchronous circuits. All possible variations are counted and lead to slow performance. Recently, configurable delay elements have been proposed to improve the performance and robustness of BD circuits, because they allow online timing adjustment based on the effective parametric variations. Akgun *et al.* reveal applying configurable delay elements in a 16-bit bundled-data ripple-carry adder yields 16% performance improvement [73]. Details of existing

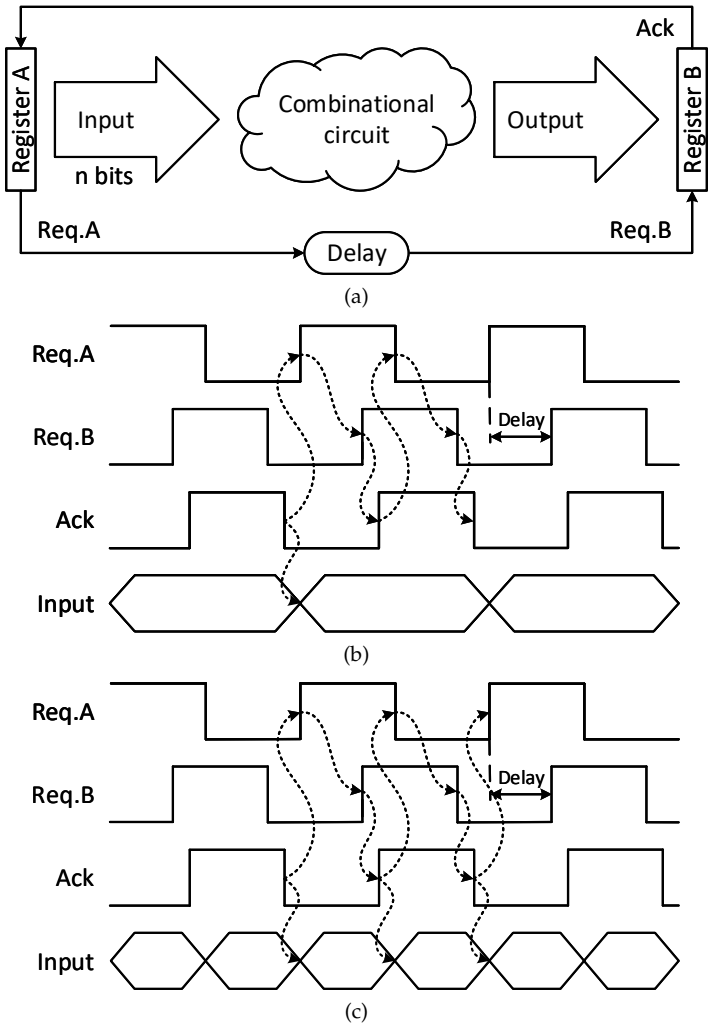


Figure 2.6: Bundled-data pipeline: (a) pipeline structure; (b) 4-phase timing diagram; (c) 2-phase timing diagram.

delay elements are given in Section 2.2 below.

Dual-rail protocol

Instead of sending a separate request signal, this protocol encodes the request signal into every bit of the data, as shown in Fig. 2.7a. Each data bit is represented by a 2-bit codeword as listed in Fig. 2.7b. For example, the input signal (In) is shown as $In.t$ and $In.f$. Therefore, we can set In to logic 0 by applying $\{In.t, In.f\} = \{0,1\}$. The 4-phase timing diagram shown in Fig. 2.7c describes data transmission. The data will be sent to the combinational circuit when RegB is ready (Ack=0). RegB will latch the data and raise the Ack signal immediately when a valid codeword is detected. Then, RegA will

fire the spacer codeword to flush the datapath. Once RegB receives the spacer, it will toggle the *Ack* signal to indicate that it is ready for the next data. As mentioned in the previous section, the 4-phase scheme is simple but provides low throughput because every consecutive data item must be separated by a spacer.

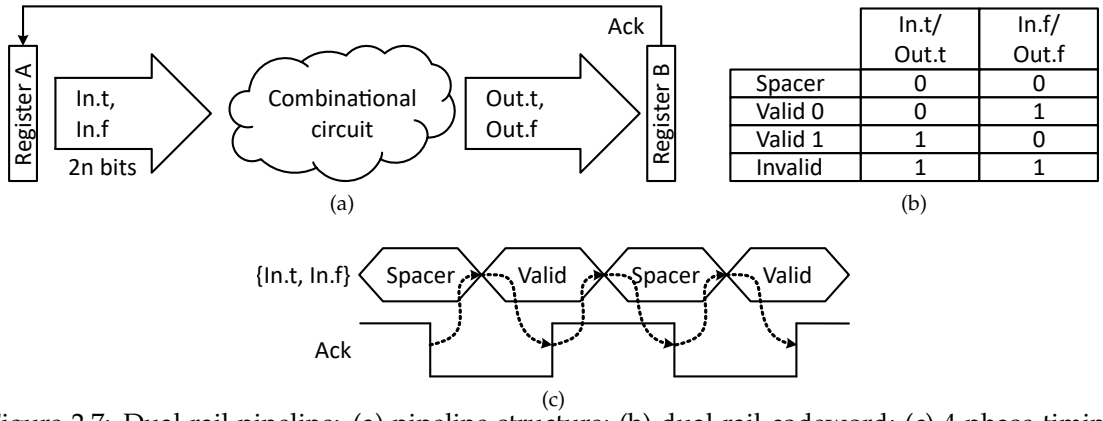


Figure 2.7: Dual-rail pipeline: (a) pipeline structure; (b) dual-rail codeword; (c) 4-phase timing diagram.

This protocol can be implemented based on a 2-phase scheme as illustrated in Fig. 2.8. The transition of f bit ($In1.f$ and $In0.f$) represents logic 0, while the transition of t bit ($In1.t$ and $In0.t$) indicates logic 1. Furthermore, one bit (t or f) of each pair ($In1$ and $In0$) must switch to represent the new data. Finally, the *Ack* signal toggles once the new data from all pairs are detected. This scheme is significantly complicated and no practical implementation of it has been proposed in the literature.

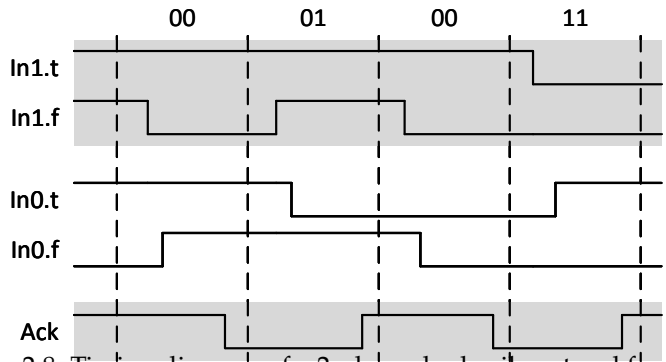


Figure 2.8: Timing diagram of a 2-phase dual-rail protocol from [19].

In summary, the DR protocol is identified as delay-insensitive because it can operate in any timing conditions without the need for configurable delay lines. Therefore, it can tolerate parametric variation. However, this protocol may consume about double

the area and power because every bit of data is represented by a 2-bit codeword. van Berkel *et al.* compare the specifications of BD and DR designs of digital compact cassette error detector. They show the BD implementation is 25% faster while consuming 33% smaller area and 50% less power [74]. Another comparison in designing a matrix-vector multiplier for discrete cosine transforms is made by Tugsinavisut *et al.* [75]. The simulation results show the BD design can save about five times the chip area and about seven times the power dissipation. Nevertheless, its throughput is approximately three times less than that of the DR design.

2.1.4 Challenges in parametric variations-tolerant design

The above design techniques have claimed improvements in performance and energy efficiency over the margin-based synchronous designs. Unfortunately, a fair comparison, where those techniques and their synchronous baseline are implemented for the same application using the same fabrication technology, has not been made in the literature.

Table 2.1 lists the aforementioned techniques and their improvements over synchronous counterparts. It can be seen that all techniques can enhance performance and power efficiency. Notice that TS's performance and power consumption is a trade-off due to the dynamic voltage scaling scheme [15]. This means the highest performance (2.0x) can be achieved when applying the highest supply voltage. On the other hand, the power dissipation is minimised (0.4x) by operating at the slowest speed. Interestingly, the DR circuit's power dissipation is low; even its size is five times larger than that of the synchronous implementation. This power reduction comes from several factors, including the data-driven nature [18], the logic synthesis algorithm [76], and the absence of clock tree and clock gating logic [77]. However, the power benefit of DR cannot be concluded as some literature; for example, Tugsinavisut *et al.* report their DR design consumes seven times higher power than its BD counterpart [75]. Considering the area requirement, the areas of PDT, TS and BD circuits increase slightly while the DR circuit size is five times larger than its synchronous counterpart. This is because, in the DR protocol, every data bit is represented by a 2-bit codeword [19].

In summary, power and area are the main limitations of IoT device developments, as mentioned earlier. Considering the recently proposed parametric variations-tolerant design techniques, we can conclude that DR is not suitable for IoT applications as its

size is enormous, whether its power figure is better or not. Although PDT, TS and BD specifications are competitive, BD offers a better opportunity for developing low-power IoT devices because it can mitigate the parametric variations while spending energy only when the data is valid; thus, there is an extremely low power loss in idle mode.

Table 2.1: Feature comparison of parametric variations-tolerant design techniques

Work	Technique	Synchronous/ Asynchronous	Performance	Power	Area
[13]	PDT	S	1.3x	0.6x	1.1x
[15]	TS	S	2.0x	0.4x	1.2x
[78]	BD	A	1.2x	0.6x	1.0x
[79]	DR	A	2.2x	0.3x	5.0x

2.2 Delay element

As technology scaling is limited by parametric variations, tunable delay elements play a crucial role in variation-tolerant design techniques, such as PDT [13, 14]. Furthermore, the circuit ageing that causes timing violations in the long term can be also addressed by self-healing techniques using tunable delays [80]. These techniques are also applicable to the mitigation of timing errors in bundled data asynchronous pipelines [17].

There are many approaches to the design of reconfigurable delay elements. Current-starved delay elements (CSDEs) have been proposed [81, 82], as depicted in Fig. 2.9a and 2.9b respectively. The delay is adjusted by controlling the charging and discharging currents of the inverter through $\{b0 - b3\}$. Because the delay generated is dependent on the sizes of the control transistors, both DEs are susceptible to process variation. Moreover, the current mirrors shown in Fig. 2.9b always draw current and consequently cause high static power consumption, which is not suitable for IoT application [84]. A simple yet effective multiplexer-based inverter-chained delay element (MuxDE) has been introduced [83], as shown in Fig. 2.9c. However, a significant amount of dynamic power is spent because of the large number of switching components. Elsewhere, a linear comparator-based design is proposed [85] which, however, is difficult to couple with digital circuits and is prone to voltage variation due to the need for an analog signal for the configuration of the delay. The above DEs also require additional storage elements, which incur the power overhead, to keep the delay configurations.

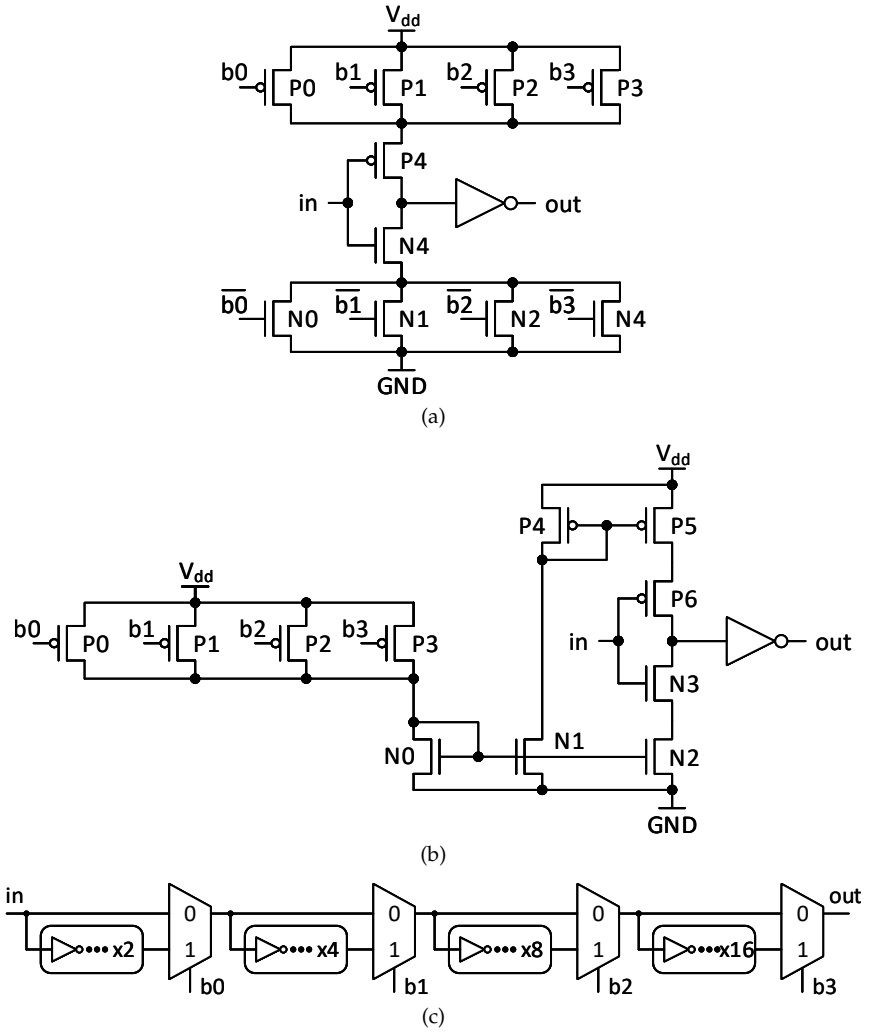


Figure 2.9: Recent DE designs: (a) CSDE from [81]; (b) CSDE from [82]; (c) MuxDE from [83].

Mohammad *et al.* reported a 2kB latch-based memory in 65nm technology consumes the power of 67.7nW [86]. Furthermore, these configurations may be lost due to the power interruptions; thus, additional power and time are needed for the delay re-initialisation when the power is back.

Several proposals to use memristor for delay elements (MemDEs) have their own limitations. For example, Fig. 2.10a shows a design which can provide the delay for one transition only [87]. Meanwhile the design in another study, as shown in Fig. 2.10b, is power-hungry due to the always-on current mirror [88], which is similar to a previous proposal [82]. The circuit in Fig. 2.10c uses a modified threshold memristor [80]. Therefore, it cannot guarantee that the memristor will be implementable. Moreover, there is a possibility that the memristor is tuned unintentionally because its threshold is

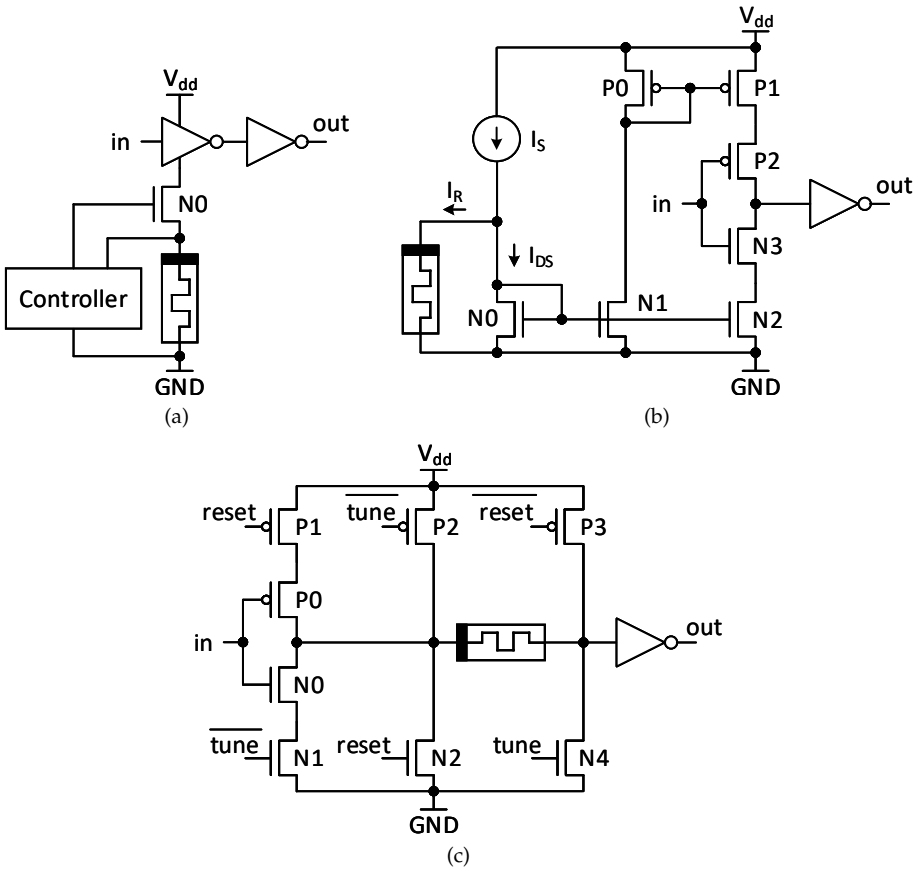


Figure 2.10: Recent MemDE designs: (a) the memristor is inserted at the bottom of the pull-down transistor [87]. This design contributes to asymmetric delay, which is not compliant with the 2-phase BD circuit requirement. The memristor voltage may exceed the threshold during the logic transition (PMOS and NMOS are on simultaneously); (b) the memristor controls the currents of P2 and P3 transistors, which results in delay tuning, through the current mirror [88]. This design is power-hungry because the current mirror is always on; (c) the memristor is attached between two inverters [80]. This design offers a highly symmetric delay power saving, but it is risky to unintended tuning, and the applied memristor parameters are impractical.

lower than the supply voltage.

Overall, building MemDEs enables support for parametric-variation tolerance by adapting the delay to match runtime conditions. The MemDE also supports power-restricted applications because the value of a delay is preserved in the memristor even when the power is off, and thus there is no need for an additional storage element. This helps to reduce overheads regarding power, time and memory space that otherwise would be inevitable for retaining and retrieving every delay value in the system. However, existing MemDEs have some drawbacks, such as unbalanced rise and fall times, high power consumption and unintended tuning. These issues are tackled in

Chapter 3.

2.3 Memristor

2.3.1 Titanium-dioxide memristor

One of the first practical memristor was invented at the HP lab in 2008 [23]. It consisted of two thin-layer titanium dioxide (TiO_2) films, one with oxygen vacancies (TiO_{2-x}), between two terminals, as shown in Fig. 2.11a. The applied power forces the oxygen vacancies towards the adjacent layer, thus reducing the memristance; these vacancies move back when the opposite power is applied. Its memristance model consists of doped and undoped regions, as illustrated in Fig. 2.11c. The doped region contains oxygen vacancies which cause low resistance, while the other region comprises pure titanium dioxide that provides high resistance. The width w of the undoped region can be changed by applying the current to move the vacancies to the other region, which causes a shift in the total memristance (R_m). Finally, when the vacancies reach the other end ($w = D$), the total memristance becomes R_{on}. In the opposite way, when all the vacancies are pushed back ($w = 0$), the total memristance becomes R_{off}. Therefore, the mathematical model is composed of these two resistors connected in series and the state variable s which denotes the ratio of the doped region width $w(t)$ to the total width D:

$$R_m = R_{on}s + R_{off}(1 - s) \tag{2.1}$$

$$s = \frac{w(t)}{D} \tag{2.2}$$

where the state variable is within the range $[0,1]$. The above equations imply that the memristance only depends on the width of the doped region. Recently, several models have been proposed to describe the rate of change of $w(t)$. The first one is the linear ion drift model [23, 89, 90] expressed in equation (2.3). However, this model does not fit most of the reported devices because their non-linear relationships are observed instead. Currently, many non-linear models are proposed, including nonlinear ion drift [91], Simmons tunnel barrier [92] and VoltageThrEshold Adaptive Memristor (VTEAM) [34].

The latter is one of the well-known models. Its background will be given in Section 2.3.3.

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

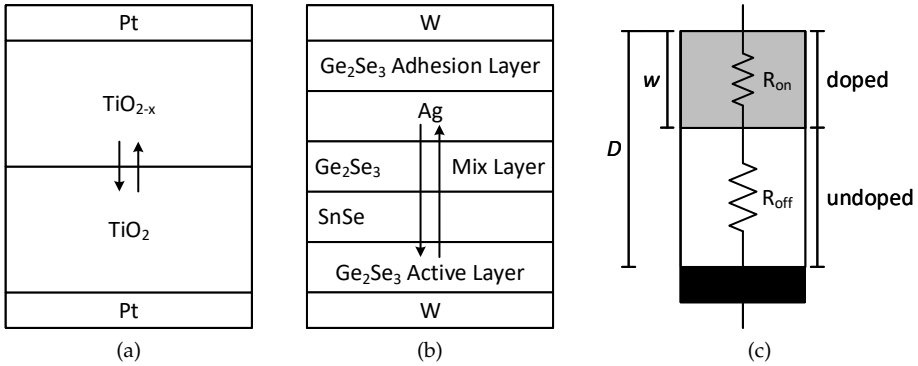


Figure 2.11: Approximate diagrams of memristive devices: (a) titanium dioxide memristor; (b) silver-chalcogenide memristor; (c) memristance model.

Titanium dioxide memristor is classified as valence change memory (VCM) due to the effect of oxygen vacancies on the valence state of the transition metal cations, which results in the change in resistivity [37]. VCM refers to the memristors that are built from metal-oxide based materials such as TiO_2, HfO_2, ZnO and Ta_2O_5.

The parametric variations also affect VCM devices. The main sources of process variations are LER and thickness fluctuation [32]. Although the former may deviate the cross-sectional area for at least 10% [93], the experimental results from Li *et al.* [33] show that R_{off} is insensitive to this variation due to the distribution of oxygen vacancies throughout the undoped region [94, 95]. Based on the report by Li *et al.*, the increased area due to the process variability likely reduce the set voltage (ON threshold) non-linearly [33]. Furthermore, they show the effect of this variation on R_{on} is approximately less than 10kΩ, which is relatively small compared to R_{off} value (≈4.5MΩ). The fluctuation in the device's thickness is caused by the imperfection of the deposition process [32]. It affects the device's width (D), resulting in the R_{on} and R_{off} variations. Abunahla *et al.* show R_{on} increases 10 times, and R_{off} is 2MΩ increasing when the thickness is grown from 30nm to 40nm [96]. The change in thickness also impacts the required length of the conductive filament, causing variability in the memristor's switching speed [31].

The change in environmental conditions such as voltage, temperature, humidity and

radiation, also affects the memristor's characteristics. Based on VTEAM model [34], the switching speed depends on the applied voltage, which may vary due to the unpredictable power source. The conductivity of the memristor's materials (metal-oxide) such as titanium dioxide and hafnium dioxide is temperature-sensitive and can be estimated by Arrhenius relation [31,35,36]. This temperature dependency is the same in chalcogenide materials in Section 2.3.2. Furthermore, the programming speed decreases non-linearly when the temperature rises [97,98]. The memristor's temperature response will be further explained and investigated in Section 2.3.4 and Chapter 4. Regarding the humidity effect, Messerschmitt *et al.* reveal the resistive switching of $Pt/SrTiO_{3-x}/Pt$ is clearly observed when the relative humidity is between 35% and 45%. Furthermore, they show the memristor's conductivity at 100% relative humidity is 4 orders of magnitude higher than that of 0% relative humidity. The radiation sensitivity of titanium dioxide memristor ($Ag/TiO_2/Cu$) is studied by Abunahla *et al.* [99]. The study shows exposing the memristor to Cs-137 662keV γ-rays can drop the switching time by 80%. The radiation effect on R_{on} and R_{off} of titanium dioxide memristor is not detectable while only R_{off} of chalcogenide phase change devices increases with respect to gamma and electron radiation exposure [100].

2.3.2 Silver-chalcogenide memristor

Currently, the silver-chalcogenide memristor [41] is the only commercially available device. It mainly comprises silver (Ag) and chalcogenide (Ge_2Se_3) layers as shown in Fig. 2.11b. Applying positive voltage forces the silver ions ($Ag+$) toward the chalcogenide active layer and results in lower memristance. Conversely, the ions can be removed by applying a negative voltage which results in higher memristance. To simulate this memristor's I-V characteristics, the fitting parameter set is available only for the VTEAM model (see Section 2.3.3 below), as listed in Table 2.3 [101]. It is notable that the models in other work [34, 102, 103] are also referred to as silver-chalcogenide memristors but their structure is different as it uses an Ag_2Se layer instead of the $SnSe$ layer, as shown in Fig. 2.11b [104]. Furthermore, this kind of memristor is no longer available in the market.

This memristor is classified as electrochemical metallisation (ECM) memristor because the memristance depends on voltage affecting the movement of silver ions in

the active area [37]. Notice that ECM may not be the only mechanism that affects the memristance. Another mechanism called phase change may involve when the applied voltage causes a sufficient temperature exceeding the glass transition temperature of the chalcogenide material. As a result, the material's structure is changed from amorphous (high R_{off}) to crystalline (low R_{off}). Note that the memristors made from Ge-rich glass, used in this thesis, are more temperature tolerance than those made from Se-rich glass. To be specific, the glass transition temperature of Ge_2Se_3 (Ge-rich glass) is ≈ 350°C [41], while the glass transition temperature of $Ge_{20}Se_{80}$ (Se-rich glass) is ≈ 160°C [105].

Campbell [41] reported the low impact of thickness variation, yet the in-depth study of the process variation effect is still unexplored. The voltage fluctuation affects the switching speed of this memristor in the same way as titanium dioxide memristor. As mentioned earlier, this memristor is robust to the phase change mechanism because it is built from Ge_2Se_3 material, which has a high glass transition temperature. The temperature responses of the chalcogenide material used to build this memristor can also be described by the Arrhenius relation [38, 39], as explained in Section 2.3.4 and Chapter 4.

2.3.3 VTEAM model

The Voltage ThrEshold Adaptive Memristor (VTEAM) model was proposed by Kvatinsky *et al.* [34]. It is a threshold-based voltage-driven model that precisely estimates all reported physical device behaviours, such as linear ion drift [23, 90], nonlinear ion drift [91] and the Simmons tunnel barrier [92]. Furthermore, VTEAM exhibits superior computation efficiency of 47.5% over the Simmons tunnel barrier model, one of the computation-intensive models [106]. Its Verilog-A implementation makes it convenient for integration with design tools.

This model utilises the memristance model in equations (2.1) and (2.2) while employing equation (2.4) to define the change rate of $w(t)$, where a_{off}, a_{on}, k_{off} and k_{on} are fitting parameters, v_{off} and v_{on} are threshold voltages and $f_{off}(w)$ and $f_{on}(w)$ are window functions to limit w within $[0, D]$ range. Although equation (2.1) is primarily introduced to describe the VCM mechanism, it can be used to estimate the ECM mechanism of silver chalcogenide memristors used in Chapter 4 because it is an behavioural model that can treat the conductive filaments as a doped region. This is confirmed by Garda and

Galias who analyse the VTEAM model parameters of silver chalcogenide memristors in Table 2.3 [101]. Equation (2.4) indicates that voltage and duration $v(t)$ are the factors that affect change in the width $w(t)$ of the doped region, which consequently changes the state variable and memristance. The memristance only shifts when the voltages across the memristor terminals $v(t)$ are greater than the v_{on} and v_{off} thresholds. As shown in Fig. 2.12, the model defines v_{on} as negative polarity while v_{off} is positive polarity. The memristance turns to R_{on} when the voltage V_{pn} less than the v_{on} is applied. Contrariwise, it turns to R_{off} when V_{pn} is higher than v_{off}. These threshold voltages are important parameters in terms of memristor properties and fabrication technology matching. To clarify this, the thresholds define the minimum operating voltage that must not be less than $Max(|v_{on}|, |v_{off}|)$.

$$\frac{dw(t)}{dt} = \begin{cases} k_{off}(\frac{v(t)}{v_{off}} - 1)^{\alpha_{off}} f_{off}(w) & ,0 < v_{off} < v \\ 0 & ,v_{on} < v < v_{off} \\ k_{on}(\frac{v(t)}{v_{on}} - 1)^{\alpha_{on}} f_{on}(w) & ,v < v_{on} < 0 \end{cases} \tag{2.4}$$

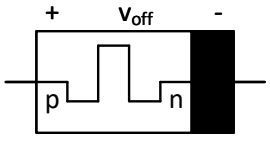


Figure 2.12: Memristor symbol with voltage threshold notations.

This model has been used in 60 publications covering 18 fabricated devices. Its parameter sets are listed in Table 2.2 and 2.3. The data in every column are collected based on the conditions that, firstly, there is no missing parameter; and secondly, they refer to physical devices. These parameters are used in determining the appropriate supply voltages and memristor types that match the target technology while meeting the requirements for switching speed and memristance range. Examples of total tuning times with different bias voltages are provided in both tables. These are useful in memristor selection as discussed in Chapter 3.

Due to its accuracy, efficiency and flexibility, this model is used in designing the delay element as described in Chapter 3. It is combined with the temperature model presented in Chapter 4 to demonstrate the temperature effect in memristance decoding in Chapter 5. To make sure that the technical issues can be observed and a prototype

circuit can be fabricated, the parameter sets in Table 2.2 and 2.3 are used without any modifications.

Table 2.2: VTEAM parameter sets

Device / Parameter	HfO_2 by Yalon *et al.* [107]	HfO_2 by Ho *et al.* [108]	TiO_2 by Strukov *et al.* [23]	TiO_2 by Yang *et al.* [24]
a_{off}	1	2	1	4
a_{on}	3	1	1	4
v_{off} (V)	500m	700m	150m	300m
v_{on} (V)	−530m	−450m	−3.5	−1.5
R_{off} (Ω)	2.5k	173.8k	10k	300k
R_{on} (Ω)	100	7k	1k	1k
k_{off} (m/s)	40.30n	28.92n	546p	91m
k_{on} (m/s)	−80	−198.72m	−73.40n	−216.2
w_{off} (nm)	10	10	10	3
IV Relation	Linear	Linear	Linear	Linear
ON time	(1.2V) 177ms	(1.2V) 678ms	(5V) 566ms	(1.8V) 53ps
OFF time	(1.2V) 62ps	(1.2V) 30ns	(5V) 318ms	(1.8V) 9ns
Obtained from	[34]	[31]	[34]	[109, 110]

Table 2.3: VTEAM parameter sets (cont.)

Device / Parameter	Ferroelectric by Chanthbouala *et al.* [111]	$Ag/SnSe/Ge_2Se_3$ by Campbell *et al.* [41]	$Ag/Ag_2Se/Ge_2Se_3$ by Oblea *et al.* [104]	Ag/gap by Pi *et al.* [112]
a_{off}	5	1	3	5
a_{on}	5	1	3	0.5
v_{off} (V)	1.4	90m	160m	400m
v_{on} (V)	−5.7	−150m	−150m	−3
R_{off} (Ω)	50M	150k	1.07k	30T
R_{on} (Ω)	150k	1.4k	387	5.4
k_{off} (m/s)	100u	11.27	2.49u	10u
k_{on} (m/s)	−30	−5.74m	−220u	−10
w_{off} (nm)	10	10^{-3}	10	35
IV relation	Linear	Linear	Linear	Linear
ON time	(7V) 98ns	(1.2V) 7fs	(1.2V) 15us	(3.3V) 175ns
OFF time	(7V) 540ns	(1.2V) 25ps	(1.2V) 133ns	(3.3V) 11ns
Obtain from	[34]	[101]	[34, 113]	[114]

2.3.4 Temperature model

Temperature fluctuations influence the characteristics of all electronic devices, including the memristor. Therefore, a temperature model is necessary for the tracking of change in the memristance and correcting decoded data. At present, most of the reported

memristor models do not include the impact of temperature [23, 34, 91, 92, 103, 106, 115–121]. A few temperature-embedded mathematical models have been presented [96, 122]; however, the authors did not provide a simulator-based model, such as in Verilog-A. Furthermore, implementing multiple differential equations, which requires several integrations, can degrade simulation efficiency and cause convergence problems [2, 98]. Although the memristance programming speed is modelled as a function of temperature in some work [97, 98], the impact of temperature on memristance is still missing. One model [90] has been proposed for the building of a memristor crossbar that can sense the circuit temperature and retain the data at the same time. Nevertheless, its linear I-V relationship does not fit practical devices, which are highly nonlinear. Some studies [31,96] report that R_{off} is highly temperature-sensitive while the impact on R_{on} is extremely low; however, that research did not cover the silver-chalcogenide memristor. One study [41] showed that silver-chalcogenide memristors can be switched at high temperature, but the impact of temperature on memristance was not reported.

Recently, the Arrhenius relation has often been used to describe the conductivity of insulating and semiconducting materials [37, 40]. The relationship is expressed in Equation (2.5), where σ is conductivity, E_a is activation energy, k_B is the Boltzmann constant and T is temperature in Kelvin. This equation is applied to the wide range of memristor-related materials such as titanium dioxide [35, 36], chalcogenide [38, 39] and hafnium dioxide [31].

$$\sigma = \sigma_0 e^{-\frac{E_a}{k_B T}} \tag{2.5}$$

In Chapter 4, this equation is integrated into the VTEAM model so as to be able to estimate the effect of temperature on memristance. The VTEAM model is selected because it has a low simulation overhead and supports a wide range of memristive devices. Temperature sensitivity can be validated by experimental data for the silver-chalcogenide memristor [41] and the data obtained from research based on titanium dioxide memristors [33, 96, 122]. This model enables further advances in research into temperature-tolerant design and temperature sensing. It is applied to the simulation of our memristance decoder in Chapter 5 to demonstrate the decoding error due to temperature and compensation for such error.

2.3.5 Memristance tuning

To precisely program memristance, two approaches have been proposed. One is to use comparators for real-time memristance monitoring [51, 54, 123–125]. In this approach, a comparator is used to continuously monitor the voltage at the middle node of a voltage divider, which is composed of a memristor and a resistor. When the voltage as read reaches the target value, either a memory controller or a related interfacing circuit stops the tuning process. This strategy yields accurate memristance tuning with a reported error of less than 1% [125]. However, the comparators always consume power and they would be more power-hungry for high-resolution data storage. Moreover, most comparators require multiple voltage references, which increases the chip area.

Another solution is to use a multiple tuning pulse width [52]. When the excitation voltage is fixed, the memristance shift depends on excitation time only, which is the pulse width in this case. In order to program memristance, therefore, an accurate control of pulse width is required. This approach is expected to reduce the amount of real-time memristance detection circuitry and consequently decrease the area used and power consumption. However, to the best of our knowledge, no excitation time model is available yet. Chapter 3 bridges this gap and delivers a methodology for realising an excitation time model for memristance tuning circuits that are based on voltage division. This concept would be applicable to a wide range of memristor circuits, including our delay element described in the same chapter.

2.3.6 Memristor's metastability

While various ongoing researches on memristors focus on their non-volatility, these memristors also exhibit a volatile or metastable characteristic. The metastability can be observed as a memristance decay shortly after applying a programming pulse, as shown in Fig. 2.13. It is caused by the relaxation of the conductive filaments formed by insufficient amplitude, width and interval of the programming pulse [126]. Cortese *et al.* reveal the filaments can also be disrupted by heat [127]. The metastability is common on both VCM and ECM devices [128], and the memristance takes time from nanoseconds [129] to seconds [116] to return to the equilibrium state. Although the metastability may affect the programming accuracy of memristor-based memory

applications, it is useful in neuromorphic computing because it mimics a biological synapse [126,130,131]. It is also utilised to prevent the sneak paths in memristive crossbar architectures [127].

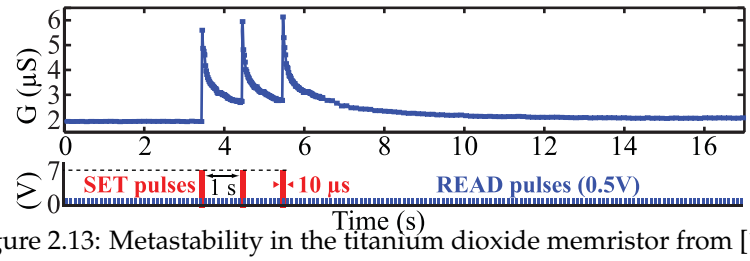


Figure 2.13: Metastability in the titanium dioxide memristor from [116].

2.4 Latch comparator

The decoder is an essential building block for the conversion of memristance to a digital value. The memristance decoding process involves iterative comparison against several reference values until the correct data is found. Therefore, a comparator is compulsory for this task because it can indicate whether the memristor's current/voltage, which reflects memristance, is greater or less than the reference value. To design a power-efficient memristance decoder, the low-power comparator is needed because it dominates the power consumption of the decoder. Recently, a latch comparator which is known as "regenerative latch" or "StrongArm", has gained attention among circuit designers due to its high sensitivity, extremely low static power and rail-to-rail outputs [132]. This section reviews the latch comparator's design, operation, offset compensation techniques and metastability issue.

The comparator circuit is shown in Fig. 2.14. It consists of the input stage N2-N3, latch stage P0-P1 and N0-N1, and finally the precharge stage P2-P5 and N4. The input stage receives differential inputs V_{in+} and V_{in-}, while the outputs are at V_{out+} and V_{out-}. The clock signal clk controls the comparator operation which occurs in two phases of precharge and evaluation. The comparison process starts from the precharge phase when $clk = 0$, and P2-P5 will connect the latch's inputs and outputs to V_{dd}. Furthermore, N4 will disconnect the circuit from ground to save power. The circuit enters the evaluation phase once $clk = 1$. The precharge transistors are OFF while N4 is ON. This allows the charges in the latch to flow to ground through N2 and N3. The current flows in the N2 and N3 arms depend on the difference between the input voltages V_{in+} and V_{in-}. Finally,

the arm that has a higher input voltage will pull its output to 0 while the output at the other arm remains at 1.

The parametric variations can cause deviations in the currents in both arms, resulting in offset. Several techniques reported in the literature can fix this offset. One is to balance the charges between the arms by connecting the capacitor arrays as in the dashed boxes in Fig. 2.14 [132]. Controlling $\{S_{0+}...S_{x+}\}$ and $\{S_{0-}...S_{x-}\}$ signals can adjust the total capacitances and balance the charges between the arms and therefore the offset is cancelled. Other solutions such as body biasing to adjust the current driving strengths of N2 and N3 [133] have been suggested, but the use of analog biasing signals is prone to runtime variations.

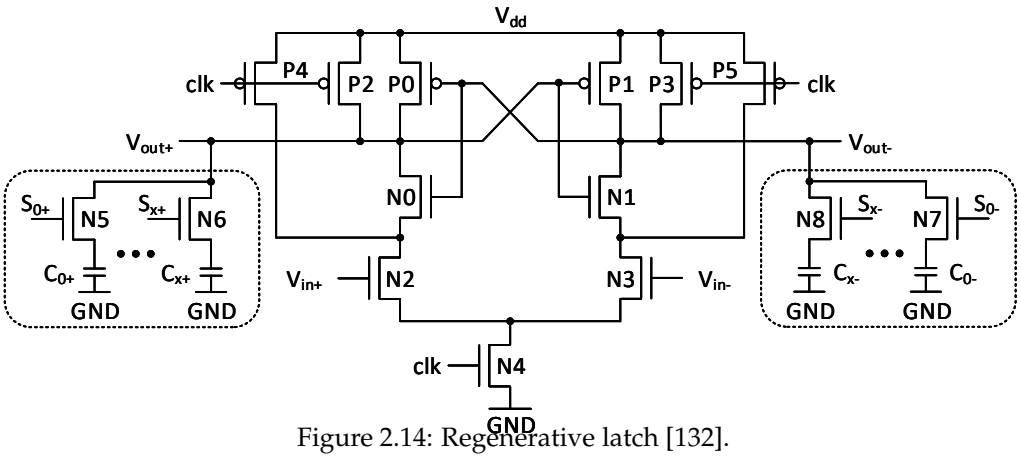


Figure 2.14: Regenerative latch [132].

2.4.1 Latch comparator's metastability

Another issue in the latch comparator is the metastability which occurs when the comparator starts to evaluate the inputs ($clk = 1$). As shown in Fig. 2.15, both output voltages are pulled to the intermediate level because both N2 and N3 are racing in drawing current. The duration of the metastable state depends on the current difference: it is longer if both currents are very close to each other and it is shorter otherwise. Then, the output voltage of the winner will drop to ground while the other returns to V_{dd}. The ambiguous state at the outputs is able to trigger the next building block and cause unexpected operation. The metastability can be filtered by applying a metastability resolver as proposed by Seitz [134].

As described in Chapter 5 the latch comparator can be modified to compare the

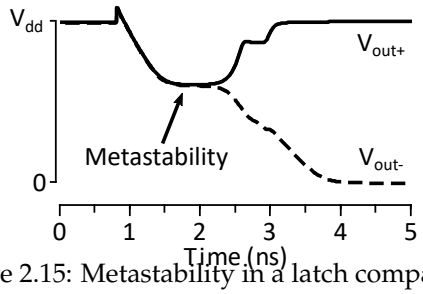


Figure 2.15: Metastability in a latch comparator.

memristor current during the decoding process. This is because a current mode circuit has intrinsic advantages over a voltage one, such as low power, wide bandwidth and lower susceptibility to power supply fluctuations [135, 136]. It features power adaptability by trading accuracy for power savings in order to survive under any energy conditions [30].

2.5 Summary

The increase in parametric variations during technology scaling obstructs the advance of IoT devices and beyond. Process variation causes deviations in the static timing of digital systems, while environmental variation and ageing affect timing dynamically. To tackle these problems, the BD asynchronous circuit is a promising solution because its performance and area are in the same range as its competitors while offering a better opportunity to save more power. Due to the use of configurable DE, BD circuits spend their power unnecessarily for retaining the delay configuration. This is not suitable for IoT applications, because this configuration will be vanished due to power interruptions, and extra time and power are needed to re-initialise delays when the power returns. Using a memristor-based delay element (MemDE) could solve the above problems, because delay information in the form of memristance can be adjusted and retained in the memristor permanently. However, existing MemDEs still have drawbacks, such as unbalanced rising and falling delays, high power consumption, and unintended tuning. These issues represent challenge (i) in the present research and are addressed in Chapter 3.

The memristor is also used as multi-bit memory device for IoT applications due to the fact that high density data can be retained without any power requirements. However, variations, especially in temperature, also affect the memristor. Temperature

significantly changes the memristor's value, called memristance, and consequently causes data errors. Therefore, an accurate temperature model is essential in order to track changes in memristance and to compensate the data accordingly. Chapter 4 uses the Arrhenius relationship to empirically model the temperature effect. The relevant parameters are extracted and the model is validated using data for the titanium dioxide memristor available in the literature and our experimental data for the silver-chalcogenide memristor, which so far is the only commercial device. Our model is integrated into the well-known VTEAM model to take advantage of the latter's accuracy, efficiency and flexibility. This chapter is the first step in improving the temperature awareness of memristive circuits, which is challenge (ii) in this study.

A memristance decoder is necessary to read data inside the memristor. Because the power sources of IoT devices are limited and unstable, the resolution of the memristance decoder must be able to scale to suit the energy available. To do this, the latch comparator is modified in Chapter 5 in order to build a resolution-scalable memristance decoder. Our decoder is based on a current-mode circuit to obtain its intrinsic advantages over a voltage-mode circuit, such as low power, wide bandwidth and lower susceptibility to power supply fluctuations. Coupled with our temperature model, the temperature awareness of our decoder is improved as it can compensate data according to changes in temperature, which is our challenge (ii).

Chapter 3

Memristor-based delay elements

Memristors are used to design two memristor-based delay elements (MemDEs) in order to improve the tolerance of parametric variations and the power efficiency of IoT devices. Section 3.1 gives an overview of system integrations where the MemDEs are applied in synchronous and asynchronous bundled-data (BD) systems. In Section 3.2, our first MemDE design is presented. This includes the top-level design where the I/O signals and state transitions are described. Then, the circuit schematic is proposed. Finally, the memristor choosing method and the unintended tuning (UT) issue are explained. Section 3.3 proposes the next version of MemDE which can prevent the UT problem. This section describes the top-level and UT-avoidance circuit design in detail.

The above designs are simplified as a General-purpose MEmristance Tuning (GMET) circuit in Section 3.4. This is done to analyse the excitation time model which is used to estimate the tuning pulse width of the transistor-memristor circuits. In Section 3.5, the simulation results of both MemDEs are shown. The circuit characterisations, such as maximum effective memristance/delay, average delay per step, minimum tuning pulse width and power consumption, are revealed. Furthermore, the pulse width estimation using the excitation time model is compared to the GMET circuit simulation. Section 3.6 then discusses the performance of the MemDEs and the excitation time model comparing to the literature. Finally, this chapter is summarised in Section 3.7.

3.1 System overview

The target of our MemDEs is to support both synchronous and asynchronous BD systems as shown in Fig. 3.1 and 3.2 respectively. To work with a synchronous system, Fig. 2.5a is modified by adding a controller and MemDE and the traditional registers are replaced by error detection registers (EDRs), which are registers that can detect timing violations. The EDR generates an error signal *err* when a timing violation is identified. Once the controller receives the *err* signal, it sends cfg and *tune* signals to increase the delay of the MemDE. The controller might integrate a counter so as to be able to decrease the delay in case the timing violation is not detected for a long period. Note that the EDR can be implemented based on circuits in the literature such as RazorFFs [15, 16], sensor flip-flops [137] and emergency detectors [80].

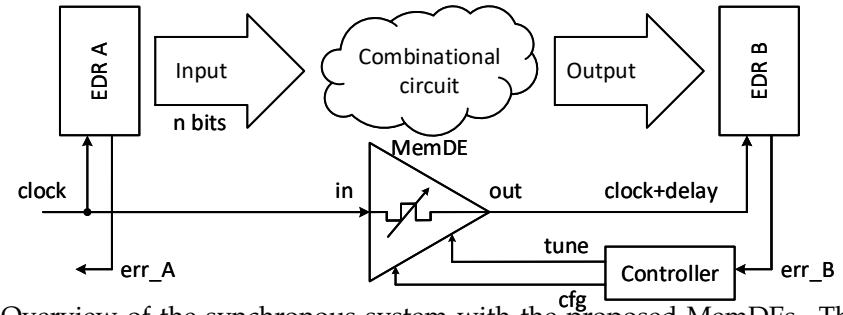


Figure 3.1: Overview of the synchronous system with the proposed MemDEs. The traditional registers are replaced by error detection registers (EDRs) that can send error signals (*err*) to the controller once timing violations are detected. Then, the controller will shift the delay of the MemDE accordingly using *cfg* and *tune* signals.

The method used to design an asynchronous BD system is similar to that of the synchronous one as illustrated in Fig. 3.2 (modified from Fig. 2.6a). All components are the same except the EDRs which need extra I/Os to support the handshaking protocol as explained in Section 2.1.3. The controller and EDRs can be implemented based on the Blade controller and error detecting latch described elsewhere [17]. Notably, the EDR and controller designs are not included in this thesis.

3.2 Memristor-based delay element design

In this section, the design of our first MemDE is explained. We start from the top-level design to describe its modes of operation and component interfaces. The circuit design

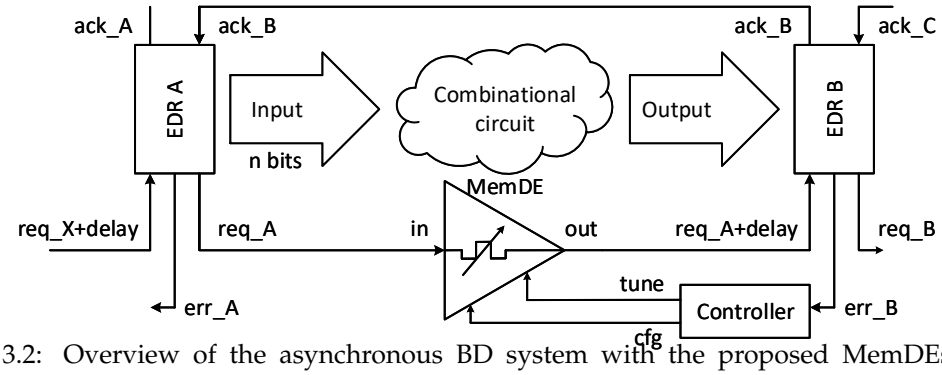


Figure 3.2: Overview of the asynchronous BD system with the proposed MemDEs. Error detection registers (EDRs) are used to replace the traditional registers to detect timing violations. Once the violation occurs, the corresponding EDR will send an error signal (*err*) to the controller. Next, the controller will send *cfg* and *tune* signals to the MemDE to shift its delay accordingly.

is then provided in detail. Next, the memristor properties are considered in the selection of an appropriate alternative. Finally, the UT problem is identified and a solution is proposed.

This MemDE requires two voltage supplies: V_{tune} for memristance tuning and V_{dd} to power the standard logic cells. In this section we use the symbols "++", "+", and "−" to represent different voltage levels, as summarised in Table 3.1. Notice that the tuning voltage must be greater than both the v_{on} and v_{off} thresholds, while the logic voltage should be lower to avoid memristance shift in normal mode. Besides, the selected tuning voltage is far lower than that of the NAND flash, which needs >20V [138].

Table 3.1: Voltage notations for MemDE

Symbol	Value (V)	Description
$++$	7	Memristance tuning voltage (V_{tune})
$+$	5	Logic 1 (V_{dd})
$-$	0	Logic 0 (Ground)

3.2.1 Top-level design

The MemDE can operate in one of three modes as summarised in Table 3.2. The transition between the operating modes is controlled by the cfg and $tune$ signals, as shown in the state diagram in Fig. 3.3.

The cfg signal selects either tuning mode (when the delay is being configured) or normal mode (when the input signal is passed through to the output with an additional delay). The $tune$ signal is used to control the tuning directions: either $tune\ up$ to gain

Table 3.2: Operation modes of the MemDE

Mode	Input			Output
	in	cfg	*tune*	*out*
Tune up (higher R)	$+/-$	$++$	$++$	$-$
Tune down (lower R)	$+/-$	$++$	$-$	$+$
Normal (pass through with delay)	$+/-$	$-$	$++/-$	$+/-$

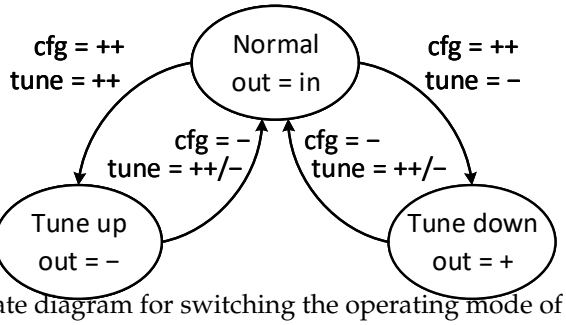


Figure 3.3: State diagram for switching the operating mode of the MemDE.

more delay or *tune down* for the opposite. The state of the *tune* signal is ignored in the normal mode and can be either "++" or "$-$".

At the top level, the proposed MemDE consists of two control inputs cfg and *tune*, a normal input *in*, a delayed output *out*, and two voltage supply pins V_{dd} and V_{tune}, as illustrated in Fig. 3.4.

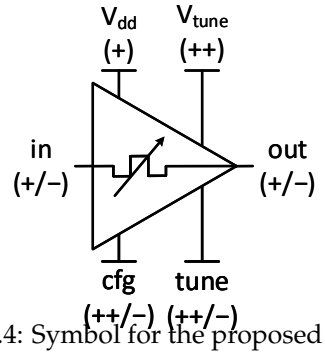


Figure 3.4: Symbol for the proposed MemDE.

3.2.2 Circuit schematic

The state diagram in Fig. 3.3 and circuit schematic in Fig. 3.5 document the circuit's operation. In the normal mode, cfg is "$-$" and turns on both P2 and N2, which form a pass gate, to pass the normal signal from the input buffer to the memristor and then to the output buffer. The state of cfg also turns off both P3 and N3 to cut the tuning network from V_{tune} and ground whether the tune is "++" or "$-$". In the tune up mode,

cfg switches to "++" and turns off the pass gate while both P3 and N3 are turned on. At the same time, the $tune$ is "++" which turns on both P5 and N4. This connects the p and n terminals of the memristor to V_{tune} and ground respectively and causes the state variable to go higher. On the other hand, in tune down mode, cfg is "++" which turns on both P3 and N3 while $tune$ changes to "$-$" and turns on P4 and N5. This also connects the memristor to V_{tune} and ground but in the opposite direction and causes the state variable to become lower. Note that the output of this circuit depends on the input voltage of the last stage buffer thus, the output is "$-$" in tune-up operation (tune = "++") and is "+" in tune-down operation (tune = "$-$").

The PMOS transistor P7, whose gate and source are connected to n and p terminals respectively, is used to deal with the UT problem as explained in Section 3.2.3. In addition, the pass gate is necessary to block the leakage current that flows from V_{tune} to V_{dd} via the body of P1 which occurs in both tuning operations. Note that the memristor can be placed in both directions depending on the thresholds. The side that has the threshold above the normal signal amplitude must face towards the pass gate to avoid a change in memristance. However, the memristor can be placed in any direction if both of its thresholds are greater than the normal signal amplitude.

All transistors except P7, which is described in Section 3.2.3, are sized to balance the rise and fall times. For high voltage AMS CMOS 0.35µm technology, the proper W_p/W_n ratio is 1. Therefore, the width of 40µm is selected for P1-P5 and N1-N5, while both P6 and N6 are sized at 20µm, which is the minimum channel width of the technology.

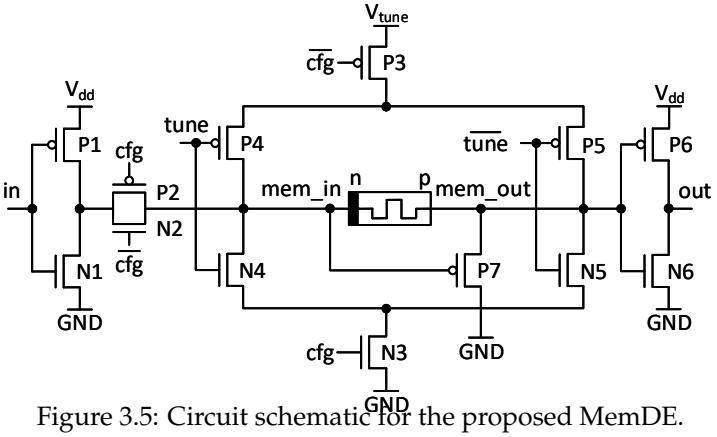


Figure 3.5: Circuit schematic for the proposed MemDE.

3.2.3 Memristor choosing and unintended tuning problem

The memristor selection criteria include the thresholds, fabrication technology and total tuning time. To prevent a shift in the state variable at normal amplitude, both memristor thresholds should be greater than V_{dd}. Meanwhile, they must also be within the operating voltage range of the target fabrication technology so as to ensure that the tuning voltage V_{tune} is appropriate and the design can be fabricated in practice. Both conditions are stated as in equations 3.1 and 3.2:

$$Min(|v_{on}|,|v_{off}|) > V_{dd} \tag{3.1}$$

$$V_{technology} > Max(|v_{on}|,|v_{off}|) \tag{3.2}$$

Unfortunately, based on the available memristors in Table 2.2 and 2.3, it is hard to find the suitable memristor. For instance, the OFF thresholds of the memristors from Strukov *et al.* [23] and Yang *et al.* [24], as shown in Tables 2.2, are too low. Thus, the signal amplitude below them is not supported by today's technology. Although this technology can provide such voltage, it is not capable of supporting the voltage above the high ON threshold at the same time. Furthermore, VTEAM is not a physics-based model; thus, we cannot modify its parameters to build a new practical device that fit our requirements. This limitation forces us to choose a memristor with only one threshold satisfied and to face the "unintended tuning problem".

The UT problem happens when at least one memristor threshold is below the signal amplitude, as illustrated in Fig. 3.6. Assume that there is a memristor with only v_{on} that has a higher threshold than V_{dd}. Connecting this side to *mem_in* and the other side to *mem_out* in Fig. 3.5 seems reasonable because the normal signal cannot exceed this high threshold. However, in normal mode, the signal at the p terminal, which is the low threshold side, is delayed and causes a voltage difference $v_p(t) - v_n(t)$ between the memristor terminals. If this voltage is larger than the OFF threshold, the state variable will shift unexpectedly. To sum up, the UT problem takes place when the conditions in equations (3.3) and (3.4) are met simultaneously. Note that if the delay is not long enough, as in the delay element presented elsewhere [80], then the UT problem may not

manifest itself.

$$V_{dd} > Min(|v_{on}|, |v_{off}|) \tag{3.3}$$

$$|v_n(t) - v_p(t)| > |v_{on}| \text{ or } |v_{off}| \tag{3.4}$$

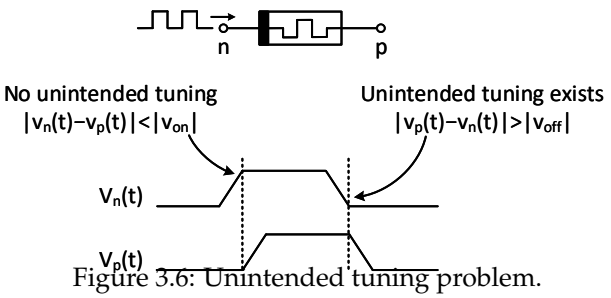


Figure 3.6: Unintended tuning problem.

To deal with this problem, the flushing transistor P7 in Fig. 3.5 is used to flush the charges at the p terminal when they are significantly greater than that at $v_n(t)$. The transistor size is selected as 100µm to maintain the growth in the state variable at a small rate. Fig. 3.7 shows the growth of the state variable with and without this transistor, where, with flushing, growth is significantly lower. However, this solution reduces the rising delay time and causes a difference between the rising and falling propagation delays, as discussed in Section 3.5.1. Furthermore, the flushing transistor also causes a high power dissipation in the tune up period because *mem_in* and *mem_out* will connect to ground and V_{tune} respectively. As a result, this transistor will conduct another current path and most of the current will be drawn through this path, as shown in Fig. 3.8. The additional path affects the voltage drop at *mem_out* as well and, hence, slows down the tune-up speed as expressed in equation (2.4).

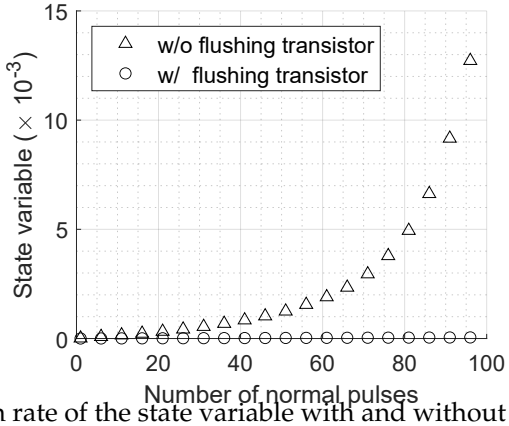


Figure 3.7: Growth rate of the state variable with and without flushing transistor.

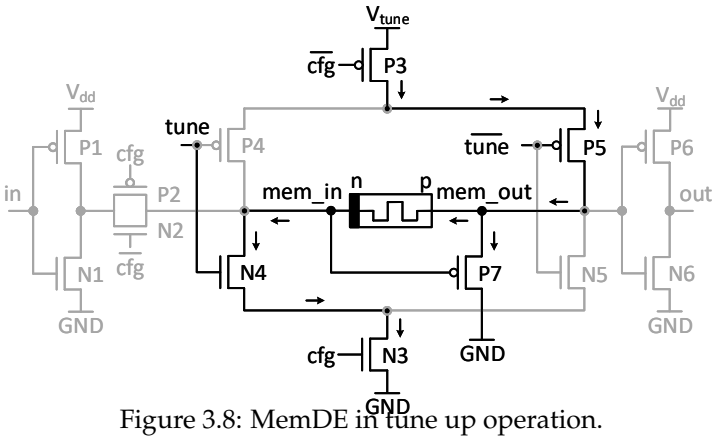


Figure 3.8: MemDE in tune up operation.

To save power, another transistor should be connected in series with the flushing one. By using cfg as a control signal, the additional transistor will be ON in normal mode and allow the usual flushing mechanism. Alternatively, it will be OFF in both tuning modes and block the current flow. This solution can reduce the tune up power consumption to the same rate as the tune down one and also decreases the tune up time because the voltage at mem_out is closer to V_{tune} than before. However, this costs an area overhead because the sizes of the flushing transistor must be doubled (200µm), plus the extra transistor's size must be equal to that of the flushing one to maintain the same path resistance. Additionally, this shorter tune up time causes a great difference in tuning interval between the tuning modes and consequently requires different tuning pulse widths, which increases the complexity of the delay control circuit (which is beyond the scope of this thesis). Therefore, to simplify the circuit operation by using a single tuning pulse width, the aforementioned transistor is ignored.

Another factor in selecting the memristor is tuning speed, which impacts the number of tuning steps. From the list of total tuning times in Tables 2.2 and 2.3, the switching of the silver-chalcogenide memristor from Campbell *et al.* [41] is the fastest (7fs for OFF to ON switching and 25ps for ON to OFF switching, based on the tuning voltage of 1.2V) and can completely turn between R_{off} and R_{on} within a single tuning step. This is useful for systems that require only two delays. If a multiple step delay is required, then the lower operating voltage or the memristors with slower tuning speed should be considered instead.

3.3 The new design with unintended tuning avoidance

This section presents the new design of the MemDE which features unintended tuning avoidance (MemDE-UTA). The main improvements compared to the previous design are:

(i) The circuit structure supports UT avoidance, and the state shift in normal mode is eliminated.

(ii) The circuit can be implemented using the smaller technology node of 180nm instead of high-voltage 0.35μm in the previous design.

(iii) The rising and falling delays are nearly symmetrical.

The information provided in this section is in line with that in the previous one; it starts from a description of the operating modes and their control logic. Then, the circuit details plus the idea of UT avoidance is presented. Note that the signals are represented in term of logic "0" and "1" instead of "++", "+" and "−" in the previous section because this design can use a single supply voltage. This is based on the assumption that the signal transition in the smaller technologies is fast enough so that the condition in equation (3.4) is not met.

3.3.1 Top-level design

The circuit still has three operating modes as listed in Table 3.3. The operating mode is controlled by the cfg and $tune$ signals as shown in Fig. 3.9. cfg is used to switch between tuning and normal modes where the input signal is transferred to the output with an additional delay. When the tuning mode is active ($cfg = 1$), the tune signal selects either tuning up ($tune = 1$) or tuning down ($tune = 0$), which result in longer and shorter delays respectively. Due to the use of a single supply voltage, the terminal for high supply voltage (V_{tune}) is omitted from the symbols as depicted in Fig. 3.10.

3.3.2 Circuit schematic

The circuit design is illustrated in Fig. 3.11. The idea to avoid UT is to arrange the connection so that V_{dd} always faces towards the n terminal of the memristor, which has

Table 3.3: Operation modes of the MemDE-UTA

Mode	Input			Output
	in	*cfg*	*tune*	*out*
Tune up (higher R)	×	1	1	0
Tune down (lower R)	×	1	0	0
Normal (pass through with delay)	0/1	0	×	0/1

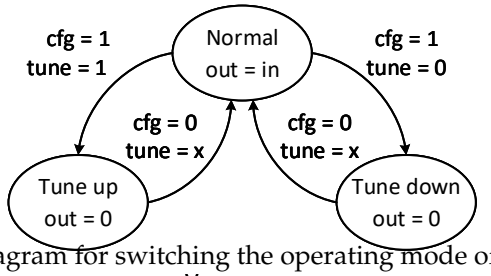


Figure 3.9: State diagram for switching the operating mode of the MemDE-UTA.

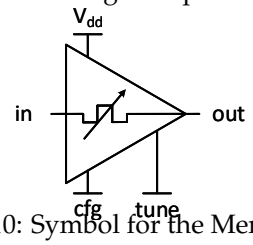


Figure 3.10: Symbol for the MemDE-UTA.

a higher threshold than the other. Considering normal operation in Fig. 3.12a, when cfg and in are 0, P1 and P2 are ON creating a current path flowing through the memristor to charge the gate capacitance of N7. The charging speed depends on the amount of current, which relies on memristance, and this charging speed contributes the delay. For instance, the charging time/delay is long if memristance is high and is short otherwise. Note that the gate of P8 is fully charged (OFF) before that of N7 (to be ON) because the memristor increases the path resistance (propagation time). Therefore, the output node out is left floating for a short period of time. Nevertheless, the effect of this floating state is not observed in our simulation in Section 3.5.2.

When in is switched to 1 (see Fig. 3.12b), the gate capacitance of P8 and N7 are discharged via the memristor and transistors N1 and N2, causing logic 1 at the output. Therefore, the rising and falling delays are approximately equal because they share the same memristance. We can see from both figures that the n terminal is re-routed to the high voltage node when in is toggled. In other words, the supply voltage is always connected to the high threshold side in both rising and falling transitions. Note that the gates of P8 and N7 will have the same voltage ($V_n = V_p$) when in has no transition.

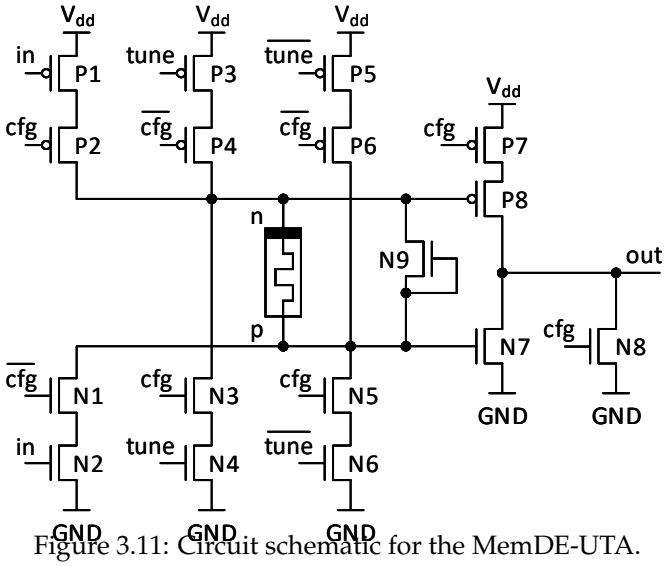


Figure 3.11: Circuit schematic for the MemDE-UTA.

Similar to the previous paragraph, *out* is floated while the P8's gate is discharging, but this floating state is not observed in Section 3.5.2.

Tuning down the memristance results in shorter delay. This can be accomplished by setting cfg and $tune$ to 1 and 0 respectively. As shown in Fig. 3.13a, this turns on P3, P4, N5 and N6, connecting the n terminal to V_{dd} and the p terminal to ground. Furthermore, the cfg signal will turn off P7 and turn on N8 to pull the output to ground. To increase the delay, both cfg and $tune$ are set to 1s. As illustrated in Fig. 3.13b, P5, P6, N3 and N4 will be ON to connect V_{dd} and ground to the other sides of the memristor. However, the tuning speed is too fast because V_{dd} is far higher than v_{off} (see Tables 2.2 and 2.3). This can be fixed by adding a diode, N9, in parallel to the memristor. This transistor can lower v_{pn} and thus slow down the tuning.

Regarding UMC 180nm technology, the transistor sizes are chosen based on a W_p/W_n ratio of 2 to balance the rise and fall times. However, the tuning transistors (P3-P6 and N3 N6) are bigger to reduce the channel resistances and increase the memristor voltage so that it is above the memristor threshold. The transistor sizes are listed in Table 3.4. Note that the minimum transistor width and length in the technology are 240nm and 180nm respectively.

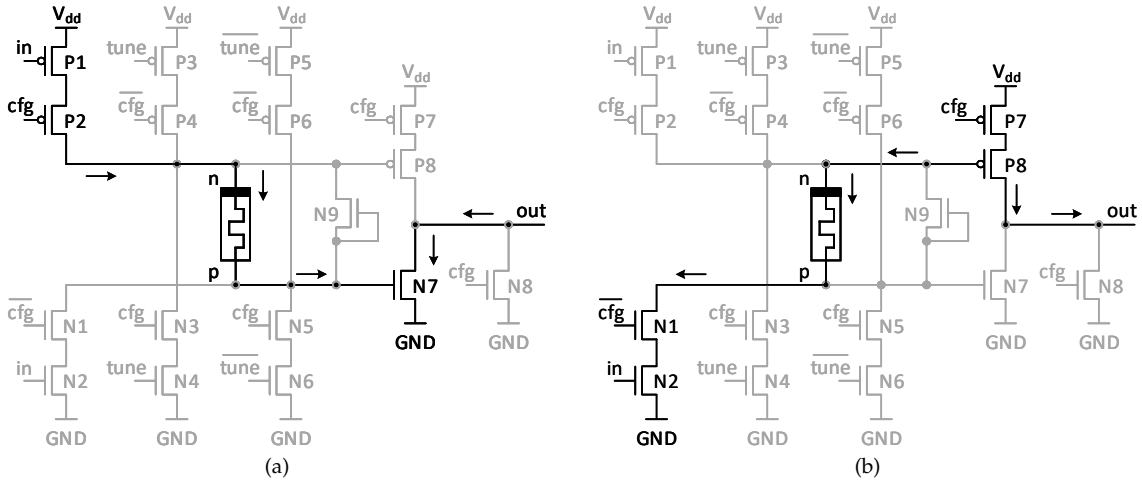


Figure 3.12: Normal operating mode of MemDE-UTA: (a) passing logic 0; (b) passing logic 1.

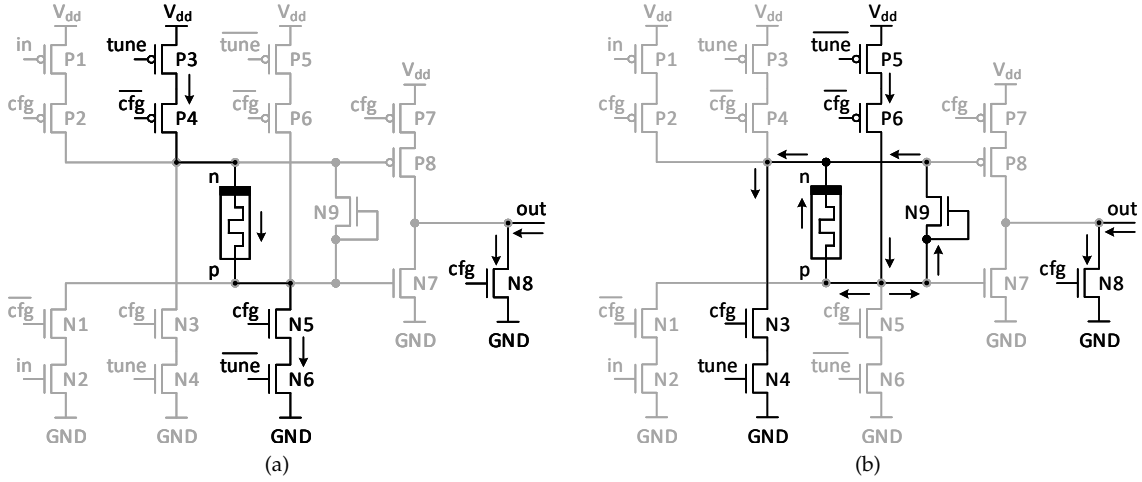


Figure 3.13: MemDE-UTA in tuning operations: (a) tune down; (b) tune up.

Table 3.4: Transistor sizes of the MemDE-UTA

Transistor	Width (nm)	Length (nm)
P1, P2, P7, P8	480	180
N1, N2, N7, N8	240	180
P3-P6, N3-N6, N9	2,880	180

3.4 Excitation time model

This section shows the simplification of the circuits in the previous sections into a general-purpose one, called the General-purpose MEmristance Tuning (GMET) circuit, to model the width of the tuning pulse. Because our model is based on a voltage divider consisting of a serially connected transistor-memristor, the model can be further applied

to other memristive circuits in a similar form, e.g. memristive crossbars. It starts at the top-level definition, which consists of the operating modes and state transitions. Then, the circuit schematic and its excitation time model are provided in detail.

3.4.1 General-purpose memristance tuning circuit

The proposed GMET circuit has three modes of operation, as shown in Table 3.5 and Fig. 3.14. It is controlled by two active-high signals, up and $down$: when both are 0s, the circuit is idle; otherwise it tunes the memristance up or down depending on the logic 1 at one of the control signals. The code where both signals are 1s is unused (invalid).

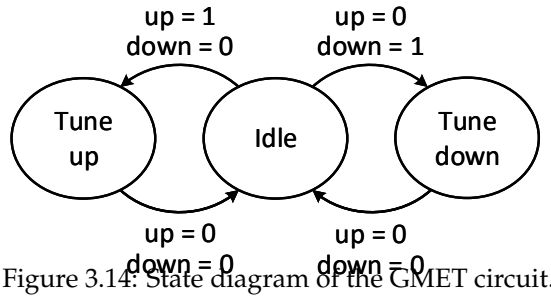


Figure 3.14: State diagram of the GMET circuit.

Table 3.5: Operating modes of the GMET circuit

Operation	Control logic	
	up	$down$
Idle	0	0
Tune down	0	1
Tune up	1	0
Invalid	1	1

3.4.2 Circuit implementation

The GMET circuit is composed of four regular MOSFETs with four control terminals, as shown in Fig. 3.15. The series connection between transistors and memristor in the figure is a common structure found in many memristive circuits such as the ones in Fig. 2.10 and both of our MemDEs in Fig. 3.5 and Fig. 3.11. The signals $\overline{up}$ and $\overline{down}$ are the counterparts of up and $down$ signals respectively. Based on AMS 0.35µm technology, all transistors are sized with a W_p/W_n ratio of 3, which gives nearly equal charge and discharge currents. In the idle mode, both control signals are 0, resulting in all transistors being OFF and memristance being preserved. The circuit enters the tune up mode when the up signal is 1, and consequently the corresponding transistors (P1 and N0)

are ON and connect the power supply and ground to the p and n terminals respectively. The voltage across these terminals induces the shift in memristance. The tune down mechanism is similar, except that the logic 1 is applied at the *down* terminal instead. Notice that using the same voltage for both tune-up and tune-down operations may cause an asymmetric tuning speed due to unequal *on* and *off* parameters i.e. $\{\alpha_{on}, k_{on}, v_{on}\}$ and $\{\alpha_{off}, k_{off}, v_{off}\}$. We may compensate this by controlling the memristor voltage in each operation through the transistor sizing (voltage divider). Therefore, the sizes of tune-up and tune-down transistor pairs ({P1, N0} and {P0, N1}) may not be identical.

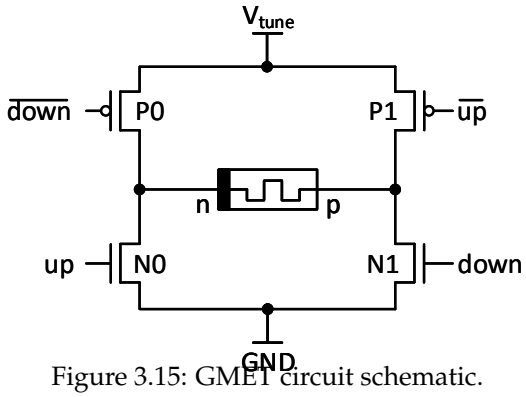


Figure 3.15: GMET circuit schematic.

3.4.3 Excitation time model

In each tuning operation, there is a current path which consists of two transistors and one memristor. The channel resistances R_p and R_n of these transistors (which are assumed to be constants in saturation mode) and memristance R_m form a voltage divider as an equivalent circuit in Fig. 3.16. Therefore, with respect to the original VTEAM model [34] with the ideal window function as in equation (2.4), the change in the length of the conductive filament $dw(t)$ depends on the excitation time dt and the voltage across the memristor $v(t)$. The latter can be replaced by a voltage divider formula, thereby giving a modified model as in equation (3.5) and its condition in equation (3.6). Here the total channel resistance R_t is constant and equal to $R_p + R_n$. For simplicity, the constants α, k and v_{th} stand for (α_{on} or α_{off}), (k_{on} or k_{off}) and (v_{on} or v_{off}), respectively. To obtain an excitation time model which describes the relationship between the amount of memristance shift ΔR_m and the excitation time Δt (tuning pulse width), the original memristance model in equation (2.1), which was proposed previously [23], is modified as expressed in equation (3.7). Then, $w(t)$ is substituted by equation (3.5), giving the final

model in equation (3.8).

$$\frac{dw(t)}{dt} = k(\frac{\frac{R_m}{R_m+R_t}v_{tune}}{v_{th}} - 1)^{\alpha} \tag{3.5}$$

$$\left|\frac{R_m}{R_m + R_t}v_{tune}\right| > |v_{th}| \tag{3.6}$$

$$\Delta R_m = \frac{R_{off} - R_{on}}{D} \cdot w(t) \tag{3.7}$$

$$\Delta R_m = \frac{R_{off} - R_{on}}{D} \cdot k(\frac{\frac{R_m}{R_m+R_t}v_{tune}}{v_{th}} - 1)^{\alpha} \cdot \Delta t \tag{3.8}$$

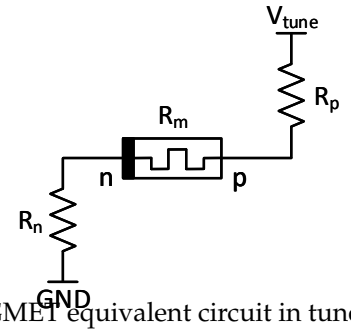


Figure 3.16: GMET equivalent circuit in tune up operation.

3.5 Simulation results

This section describes three experiments conducted on both the MemDE and MemDE-UTA to achieve the following goals:

- To identify the minimum and maximum effective memristances/delays; i.e. to find the lower and upper limits of the circuit.
- To examine the minimum tuning pulse that works well across the memristance range, and the average delay per step.
- To measure power consumption in different operating modes.

Furthermore, the GMET circuit is simulated to validate the accuracy of the excitation time model.

3.5.1 Memristor-based delay element

Experiments were conducted using Cadence Spectre with high voltage AMS CMOS 0.35µm technology and the VTEAM memristor model [34]. Biolek window function [139] is chosen because this function allows the state variable to change when its value reaches either 0 or 1 boundary [106]. Furthermore, Slipko and Pershin demonstrated Biolek is a window function that converges the state variable to a fixed value when applying periodic alternating polarity pulses [140]. Therefore, this window function is useful for the pulse-based tuning and spiking neural network applications because the pulse responses are consistent. The fitting parameters for a ferroelectric memristor from Table 2.3 were chosen because of its wide memristance range, moderate tuning time (in nanoseconds) and ON threshold that suits the operating voltage (5V and above). Using this memristor, the V_{tune} was set above the highest threshold of 7V while V_{dd} was set at 5V to let the transistors operate correctly. Regarding a very high memristance of the selected memristor, ten identical devices were connected in parallel, resulting in 15kΩ and 5MΩ respectively as the actual minimum and maximum memristances. The normal signal frequency in all experiments was set to 10MHz.

Minimum and Maximum effective memristances and their delays

The whole memristance range is not usable because excessive memristance causes an insufficient charging current at mem_out, preventing V_{mem_out} from reaching V_{dd} when in is "$-$", while mem_out is discharged quickly once in is "$+$" due to the large flushing transistor. This leads to the falling delay saturation, while the rising delay is almost flat. Consequently, a delay saturation will be observed when averaging both rising and falling delays. On the other hand, increasing the memristance can trigger unintended tuning, as explained in Section 3.2.3. Totally, the maximum memristance is considered as the highest value where neither delay saturation nor unintended tuning occurs.

The maximum memristance can be observed by applying tuning pulses (3ns pulse width) in tune up mode until the delay saturates. From the simulation results in Fig. 3.17, the average delay grows non-linearly and saturates at the 6^{th} pulse, which indicates a maximum delay of 13.54ns and a state variable value of 40×10^{-3}. This value can be converted to the maximum effective memristance of 214kΩ using equations (2.1) and (2.2).

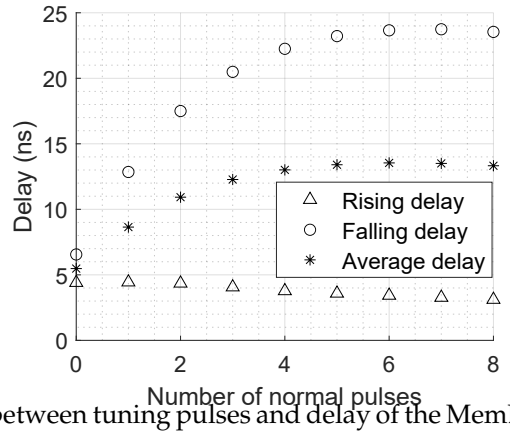


Figure 3.17: Relationship between tuning pulses and delay of the MemDE (initial state variable= 0, initial memristance = 15kΩ, pulse width = 3ns). The falling delay increases due to the increasing memristance and saturates at the 6^{th} pulse. This is not the case for the rising delay because the charges at *mem_out* are wiped through P7 instead of the memristor. The discharging speed is faster when the memristance increases.

According to the effect of the flushing transistor, the rising delay remains low because the charges are flushed quickly. The large difference between the rise and fall times still exists even the widths of N1 and P6, which relate to the rise time, are decreased. In addition, the minimum delay is measured as 5.48ns and the average delay per step are obtained as 1.34ns.

Minimum tuning pulse width

To find the minimum tuning pulse width, a simulation was run by sweeping the tuning pulse width in tune up mode from 1ns to 5ns with a 1ns increment per step. The waveforms in Fig. 3.18 indicate that the state variable started to increase at a pulse width of 2ns. In tune down mode, the state variable was initialised to the maximum effective memristance from the previous section. The simulation results in Fig. 3.19 show the minimum pulse width of 3ns instead. The variation in pulse width comes from the difference in memristances, where the lower one allows the signal to swing faster. In order to use the same pulse width for the whole range, the 3ns pulse was assigned as the minimum tuning pulse width for all experiments.

It is notable that the glitches in the *out* signal in Fig. 3.18 were initially ignored, as the solutions depend on system implementation strategies. For instance, it can be adopted with a clocked circuit without modification, since the glitches that occur between the clock edges do not affect the flip-flop [80]. Alternatively, the last stage buffer can be replaced by a tri-state one, which is useful for asynchronous circuit implementations.

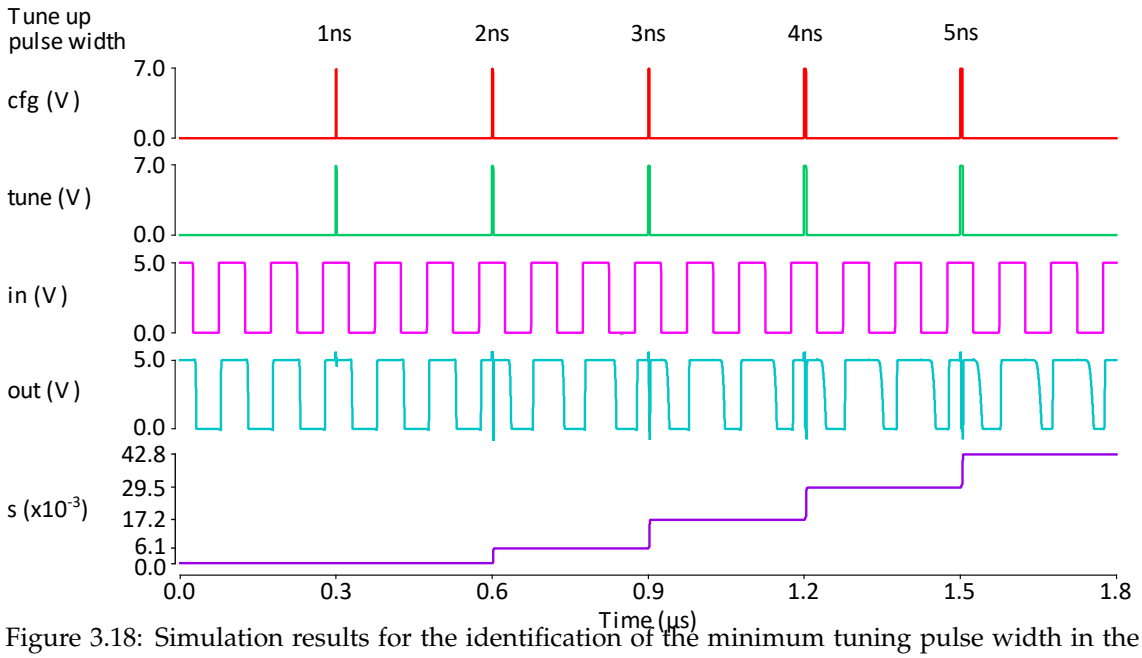


Figure 3.18: Simulation results for the identification of the minimum tuning pulse width in the tune up mode of the MemDE.

Additionally, from the *out* signal in Fig. 3.19, the first output after the tune down operation was missing due to the fact that all internal charges at *mem_out* were wiped during the tuning through N5 (cfg = "++" and tune = "−") and could not be restored in time. A solution to this issue is the subject of future work.

Power consumption

Values of static and dynamic power consumption are measured separately. The static power of 14pW is observed as an average value from all combinations of the DC input signals (*in* and *tune*). The dynamic power of 203µW is measured when a pulse train is applied at *in*. Note that we also use this measurement method for the MuxDE discussed in Section 3.6.1.

The energy per step for tune up and tune down operations is measured when the 3ns-width pulses are sent to both cfg and *tune* terminals. Energy-per-step readings of 42pJ and 1.4pJ are observed as the results of tune up and tune down respectively. The tune up energy per step is significantly higher due to the flushing transistor, as explained in Section 3.2.3. Although the tune down energy per step is approximately 40 times lower, it may require more pulses due to the slower speed of memristance shifting, thus resulting in a higher energy per delay unit tuned. Our MemDE may consume high

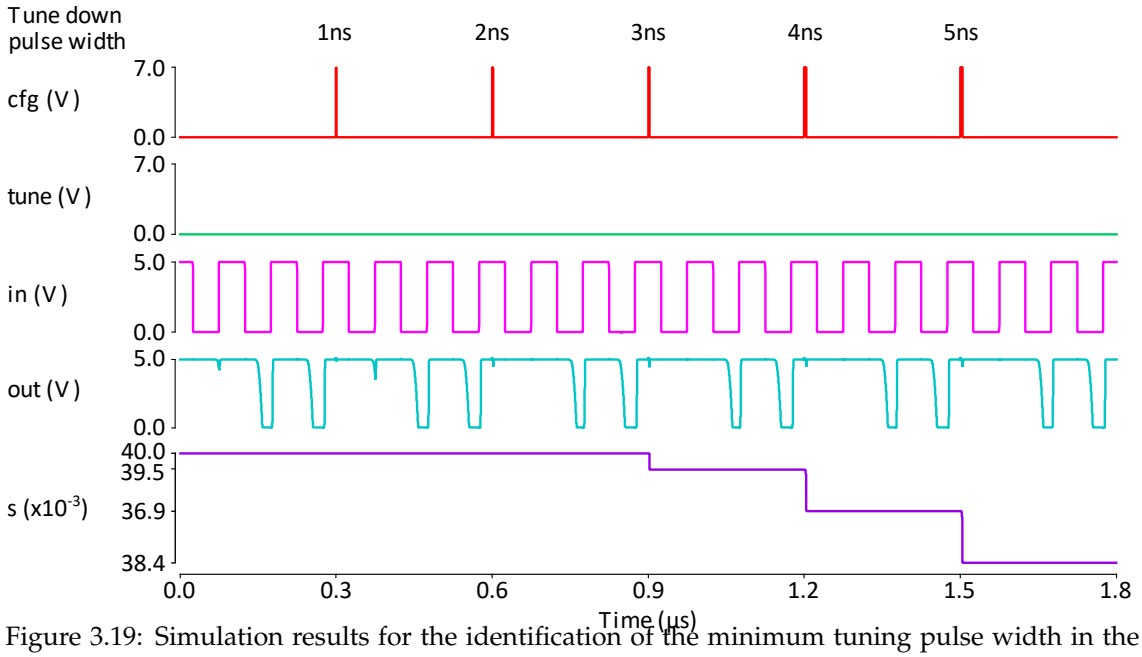


Figure 3.19: Simulation results for the identification of the minimum tuning pulse width in the tune down mode of the MemDE.

tuning energy, but it has a significantly low power consumption in normal mode, as shown in Section 3.6.1.

3.5.2 Memristor-based delay element with unintended tuning avoidance

The simulations in this section are based on UMC 180nm technology and the VTEAM model [34]. The parameter set for a titanium dioxide memristor by Yang *et al.* in Table 2.2 [109, 110] is selected because its threshold of 1.5V is within the operating range of the technology, which is 1.8V. Although the operating voltage is higher than the threshold, the signal transition in normal mode is fast enough to keep the memristor voltage below the threshold. This section repeats the procedures in the previous section to test the circuit's functionality and to characterise parameters such as minimum and maximum delays, minimum tuning pulse width and power consumption. Furthermore, the memristance range of 1kΩ - 300kΩ is already suitable for our application, and hence there is no need for parallel memristors as in the previous design. The frequency of the input signal in all simulations was set to 100MHz.

Minimum and Maximum effective memristances and their delays

The maximum memristance is considered based on the same conditions: it is the maximum value that is clear from delay saturation and UT. To find the maximum memristance, a number of 300ps tuning pulses are applied until one of the above conditions exists. In this simulation, our DE experiences the UT before the delay saturation. The relationships between the number of tuning pulses and delay are shown as circle markers in Figs. 3.20a and 3.20b. Note that only data before the presence of UT are displayed in the figures. The first figure shows a maximum delay of 1.44ns, which is a result of 167kΩ memristance (state variable = 556×10^{-3}) observed at the 25^{th} pulse. The delay in tune down mode in Fig. 3.20b is non-linear and saturates at the 47^{th} pulse where the minimum delay is 55ps, which is from 16kΩ memristance (state variable = 51×10^{-3}). This delay saturation happens because of the effect of the voltage divider between the channel resistance and memristance, as demonstrated in Section 3.5.3. The growth of the tune up delay in Fig. 3.20a is different from that in Fig. 3.17 because the delay in the previous design is dominated by internal signal distortion, which is not the case in this design. We can use the above data to calculate the average delay per step for tune up and tune down operations as 36ps and 19ps respectively. The graphs for the pulse widths of 200ps and 400ps are discussed in the next section.

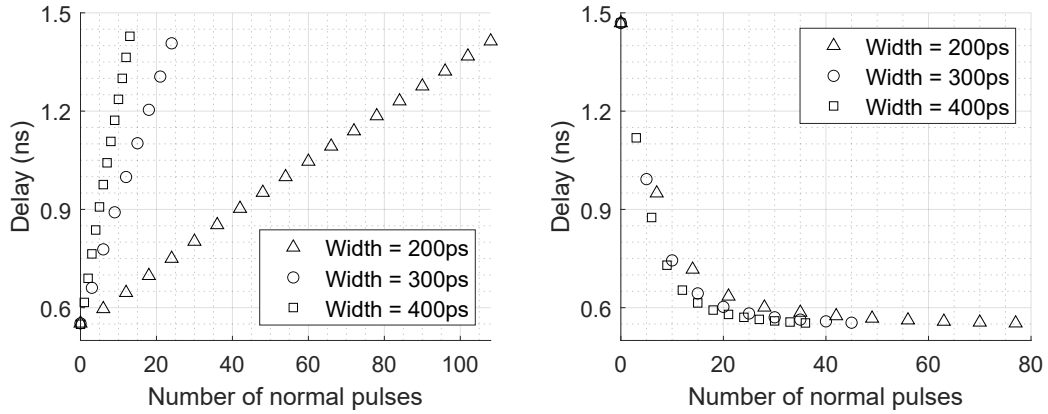


Figure 3.20: Relationship between tuning pulse and delay of the MemDE-UTA before UT occurs: (a) tune up operation (initial state variable = 51×10^{-3}, initial memristance = 16kΩ); (b) tune down operation (initial state variable = 556×10^{-3}, initial memristance = 167kΩ)

Minimum tuning pulse width

The procedures in Section 3.5.1 are repeated to find the minimum pulse width that can shift the memristance. To determine this value for the tune up operation, the state variable is initialised to 51×10^{-3}, which is the minimum value from the previous simulation. Then, two identical signals, which are series of pulses ranging from 100ps to 500ps with 100ps increments, are applied as cfg and *tune* signals in Fig. 3.21. From the figure, the state variable starts to shift when the pulse width is 200ps. Additionally, the tuning pulse of 100ps causes a spike at the *out* signal because the relevant transistors are not switched properly.

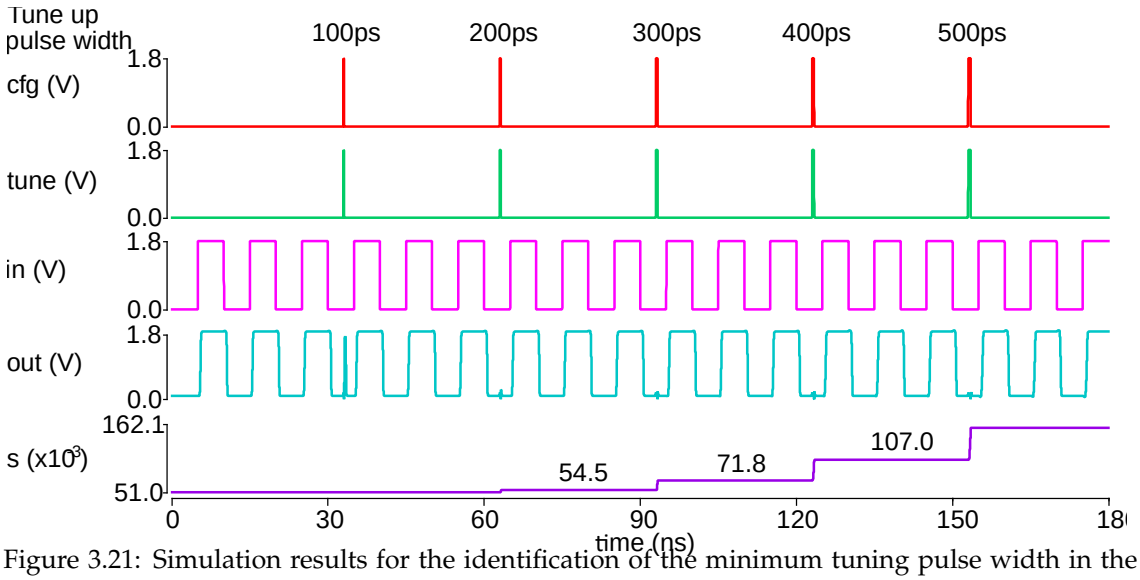


Figure 3.21: Simulation results for the identification of the minimum tuning pulse width in the tune up mode of the MemDE-UTA.

The state variable is set to 556×10^{-3} and the same pattern of the cfg signal is applied for the tune down operation as depicted in Fig. 3.22. Note that the *tune* signal is 0 at this time. Unlike in the tune up operation, the memristance can be shifted even at the pulse width of 100ps. This is because the high memristance can quickly create sufficient memristor voltage. To be able to tune the delay in both directions, we can choose 200ps as the minimum tuning pulse width.

It can be noted that, depending on design constraints, a longer pulse width can be chosen for faster tuning speed. For example, to replace a 4-bit MuxDE, which has 16 delay steps, a MemDE-UTA with 200ps tuning pulse is not a good choice because it has 111 delay steps (tuning pulses, see Fig. 3.20a), which is too high compared to that of the

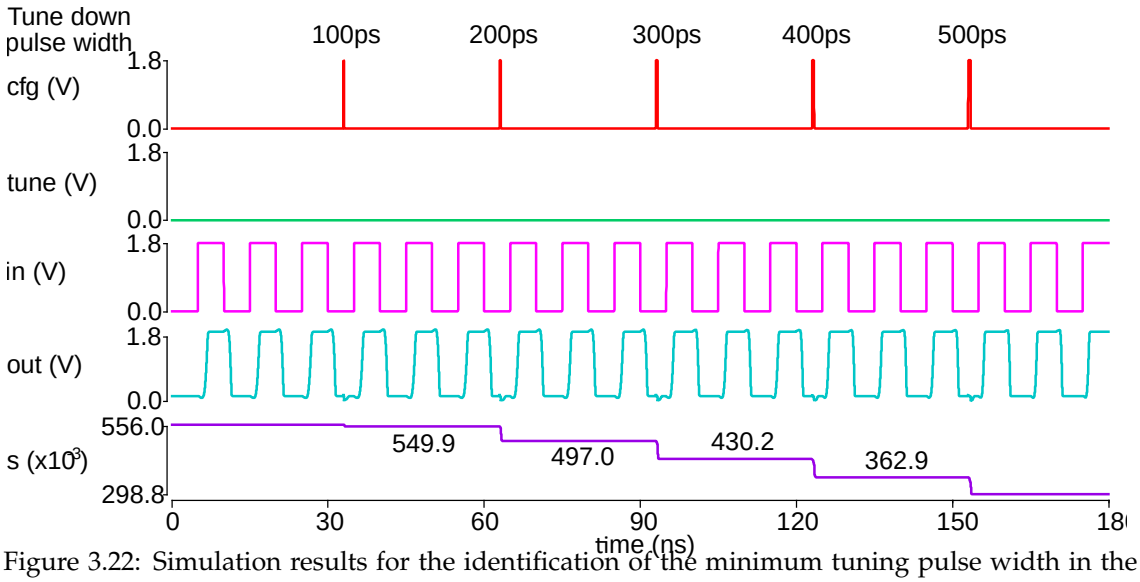


Figure 3.22: Simulation results for the identification of the minimum tuning pulse width in the tune down mode of the MemDE-UTA.

MuxDE. A better choice would be the MemDE-UTA with a tuning pulse of 300ps because it has slightly more delay steps (25 pulses, see Fig. 3.20a) in shifting the delay from the minimum value to the maximum.

Power consumption

Regarding static power measurement, DC voltages are applied at the inputs of the circuit. cfg is set as 0 to operate in normal mode, while in and $tune$ are set to all possible combinations. Based on these configurations, the static power of 230pW is measured. The dynamic power of 7.52µW is measured when a pulse train is applied at in. Note that we also use this measurement method for the MuxDE discussed in Section 3.6.1.

To measure both tune up energy and tune down energy, a tuning pulse of 300ps with a 10ps signal transition is applied to cfg and $tune$. The energy is measured for 320ps duration to cover the pulse and its transition. The tune-up and -down energies of 180fJ and 80fJ are measured respectively. The tune-up energy per step is higher because the bypass transistor N9 is active and reduces path resistance.

Unintended tuning avoidance

The in terminal is fed by 500 pulses while the circuit is operated in normal mode to observe UT over the long term. The initial state variable is set to 556×10^{-3}, which is

the maximum value before UT occurrence as demonstrated in the above section. The simulation results in Fig. 3.23 confirm that our design is free from UT, since the state variable can be plotted as a horizontal line. This result outperforms that of the previous MemDE where the UT problem still exists even when the flushing transistor is attached. Note that, in the physical implementation, the state variable may decay even the UT is prevented. This is because of the volatile effects, which the VTEAM model does not account for.

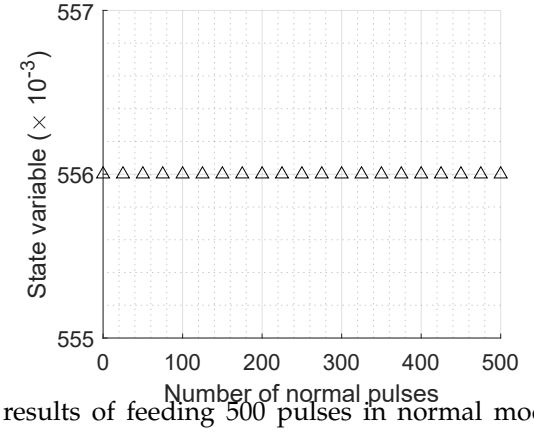


Figure 3.23: Simulation results of feeding 500 pulses in normal mode operation; initial state variable = 556×10^{-3}.

3.5.3 Excitation time model

To evaluate the accuracy of the proposed model, a memristor tuning experiment was conducted. All simulations are based on AMS CMOS 0.35µm technology and the VTEAM memristor model [34] with an ideal window function. The parameter set of Strukov *et al.*'s memristor in Table 2.2 is applied. Note that we assume that the excitation time is a product of the pulse count and width, where multiple pulses represent a longer excitation time.

This section is divided into three parts: The first part considers the different behaviour of an isolated memristor and a GMET circuit. The average value of R_t corresponding to each selected pulse width is also obtained in this part. In the second part, the proposed model (equation (3.8)) and its condition (equation (3.6)) are evaluated against the simulation data for the GMET circuit. Then, the saturation points are compared in the third part. In the last part, the issue in applying the model to the smaller technology is revealed.

GMET circuit and the single memristor

In this part, 5V amplitude pulses with a pulse width range from 20ms to 100ms (in 20ms increments) are applied to both the single memristor and the GMET circuit until memristance saturates. Then, the memristance shift corresponding to each input pulse is observed. This experiment is performed in both tune up and tune down operations. An example of the results observed is illustrated in the graphs in Fig. 3.24. These reveal the relationship between memristance shift and the pulse counts of the single memristor and the GMET circuit with 40ms pulse width.

The tune up graphs in Fig. 3.24a show that the change in memristance in the GMET circuit is slower than that of the single memristor because the voltage divider decreases the voltage across the memristor. The slope of the GMET graph is small from the beginning due to the increase in memristance. Then, the plot becomes near-linear as memristance finally dominates the voltage divider. Note that the memristance of the single memristor rises linearly throughout the simulation because there is no component to share the voltage.

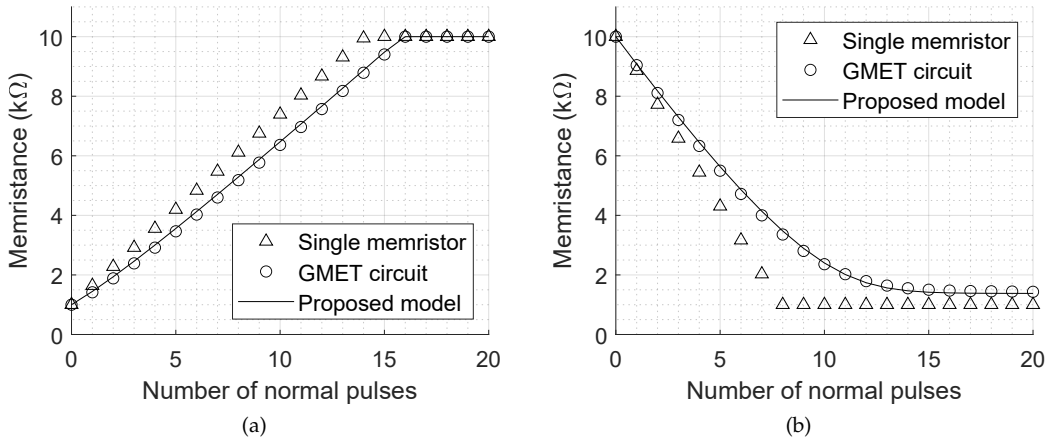


Figure 3.24: Simulation results of memristance vs number of tuning pulses based on AMS 0.35µm technology with the VTEAM model and parameters of Strukov *et al.*'s memristor, with pulse amplitude V_{tune}= 5V, and pulse width t = 40ms: (a) tune up operation; (b) tune down operation.

Regarding the tune down operation (Fig. 3.24b), the memristance shift of the GMET circuit decreases non-linearly due to the voltage divider effect. When the memristance goes low, the channel resistance of both tuning transistors becomes significant and reduces the voltage drop across the memristor. When the memristor voltage is smaller than one of its thresholds, the memristance saturates and never reaches R_{on}. In this case,

the saturation point is approximately at 1.5kΩ. Both non-linear and early-saturation issues can be solved by using the memristor with a higher R_{on}, which can be achieved by device sizing [33] such that it can always dominate the voltage divider. In addition, to avoid these issues, the supply voltage should be considered as a part of the voltage divider that directly controls the memristor voltage. Therefore, using sufficiently high voltage is an alternative solution, but it is power-intensive and limited by fabrication technology. Without the voltage divider as in the GMET circuit, the memristance of the single memristor reduces linearly until it saturates at R_{on}. The behaviour of both tune up and tune down operations is similar for any specific pulse width. The average values of R_t, one for each graph, are listed in Table 3.6.

Table 3.6: Average channel resistances obtained from pulse width sweeping

Operation	Pulse width (ms)	Average R_t (Ω)	Error (%)	
			Avg.	Max
Tune up	20	289	4.36	7.00
	40	407	1.97	3.01
	60	430	0.98	1.43
	80	431	0.32	0.47
	100	429	0.34	0.79
Tune down	20	548	5.49	10.26
	40	591	3.17	5.64
	60	600	3.38	8.58
	80	604	3.75	13.25
	100	648	3.53	11.06

GMET circuit and its model

Each R_t from the previous section is applied to the proposed model in equation (3.8) which is used to calculate the shifts in memristance. For example, based on a 40ms tuning pulse width, the graphs for the proposed model are compared with those for the GMET circuit simulations as depicted in Fig. 3.24. It can be seen that the calculated data are close to those from the simulation. The differences are listed as error in Table 3.6, and the maximum and average error of the tune up operation decline when the pulse width increases while both errors are independent of pulse-width in the tune down operation.

High error always occurs when the memristance is close to R_{on}. However, the maximum error for all operations is less than 7% when memristance is above 2.5kΩ. This is because the actual transistor channel resistance is very different from the average one

and it also dominates the voltage divider. Therefore, working at higher memristance is recommended. The values of percentage error mentioned are compared with the results of other research in Section 3.6.2.

Saturation points

The pulse counts to reach the saturation points for the applied pulse widths are illustrated in Fig. 3.25. It is clear that the transistor channel resistance affects the difference in memristance shift between that of the device and of the GMET circuit. This difference is large enough to shift the saturation point of the GMET circuit. Note that the channel resistance effect is common in serially connected memristor-transistor circuits, e.g. 1T1R structures in [43, 51, 141]; hence, our method can be applied to those circuits. Regarding our model, the predicted saturation points mostly match those of the GMET circuit. All mismatches between the GMET circuit and the model arise during operation at below 2.5kΩ memristance. Furthermore, these mismatches are consistent with the maximum errors listed in Table 3.6 because they occur at the pulse widths that have high values of maximum error (20ms for tune up and 20ms, 80ms and 100ms for tune down operations). Therefore, working at higher memristance can provide identical graphs. Note that the error values for 40ms and 60ms pulse widths are very small, so that the saturation points can be reached with the same pulse counts.

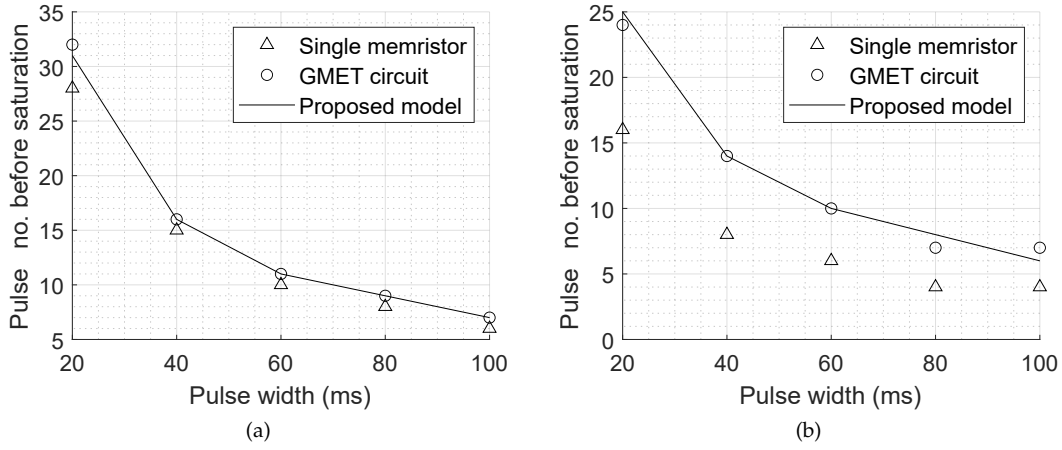


Figure 3.25: Number of pulses before saturation vs pulse width, with pulse amplitude V_{tune} = 5V and pulse width t = 40ms: (a) tune up operation; (b) tune down operation.

Accuracy loss in the smaller technology

Although the estimation is accurate for 0.35µm technology, this is not the case for 180nm technology, as shown in Fig. 3.26a. In tune down operations, the model's calculation (solid line) is very different from the simulation (circle markers). This is due to the spikes in the v_n and v_p signals of both memristor terminals, as illustrated in Fig. 3.26b. These signals cause an over-tuning and result in faster saturation as indicated in Fig. 3.26a.

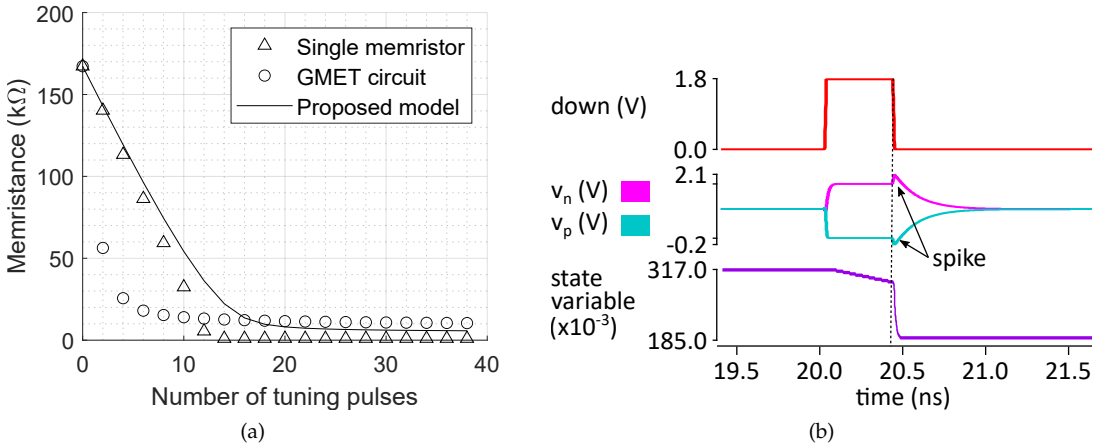


Figure 3.26: Memristance vs number of tuning pulses based on UMC 180nm technology with the VTEAM model and the parameters from prior research [109, 110], where pulse amplitude V_{tune}= 1.8V, pulse width t = 400ps: (a) tune down operation; (b) the spike at the falling transition of the *down* signal which causes overtuning.

3.6 Discussion

3.6.1 Memristor-based delay element

Table 3.7 gives an overview of the reported results from recent work and compares them against the proposed solutions.

The CSDE proposed by Maymandi-Nejad and Sachdev [82] can range in 360ps with 5-bit parallel control. This circuit consumes more power in normal mode, even though the technology is smaller, because the current mirror always connects the power source to the ground. Moreover, this solution is vulnerable to process variations as it depends on the precise sizing of the transistors.

Tschantz *et al.*'s MuxDE [83] is replicated in two technologies, the AMS 0.35µm HV and UMC 180nm, to give a fair comparison with both of our MemDE designs.

Table 3.7: Comparison of the specifications of the proposed MemDEs

Work	Delay min. (max.) (s)	Step time (s)	Normal mode power (W)		Tuning energy (J)	Tr. no.	Area (μm^2)	Tech. & model (m)
			Static	Dyn.				
[82]	2.06n (2.42n)	$\geq$ 2p	340μ max.	N/A	N/A	16	5,000	180n
3b-MuxDE [83]	7.1n (15.6n)	1.06n	3p	666μ	N/A	70	74,800	0.35μ HV
4b-MuxDE [83]	0.5n (1.5n)	63p	1.16n	100μ	N/A	108	774	180n
[85]	0.5n (5.5n)	4.2n	50μ	N/A	N/A	18	N/A	180n
[88]	30n (3u)	N/A	N/A	N/A	N/A	7	N/A	N/A + Biolek
[87]	809p (822p)	N/A	N/A	N/A	N/A	5	N/A	40n + mod. Biolek
[80]	N/A (140p)	N/A	N/A	N/A	N/A	14	0.95	45n + mod. Yakopcic
MemDE	5.48n (13.54n)	1.34n	14p	203μ	42p (1.4p)	17	30,800	0.35μ HV + VTEAM
MemDE UTA	0.55n (1.44n)	36p (19p)	1.39n	7.5μ	180f (80f)	21	560	180n + VTEAM

The control bits of the one with 0.35µm technology is selected as 3-bit while 4-bit is selected for the other. This is to provide similar maximum delays to their competitors. They consume higher dynamic power due to the higher numbers of logic gates. Their comparisons to our circuits are discussed at the end of the section.

The circuit designed by Saraj *et al.* [85] provides accurate delay control with the smallest static power dissipation. By using a comparator-based design, the delay shifts linearly with lower power consumption than the CSDE. However, it needs analog control which makes it not suitable for digital applications.

Meanwhile Mokhtar and Abdullah's circuit [88] exhibits the widest range at 2.97µs. Nevertheless, it uses a memristor as part of a current mirror, so there is still a current path that always draws energy, which is the same as with Li *et al.*'s [82]. Although this work reports the fewest transistors, this number does not include the transistors for the memristor interface.

Zhang *et al.*'s design [87] yields a short delay range at 13ps due to the narrow memristance boundary. By connecting a memristor in series with the pull-down path, the delay happens only on the rising transition, so another memristor and memristance

matching are required in order to provide the same delay as for the falling transitions. Furthermore, the voltage divider structure limits the maximum memristance and thus the achievable delay. This happens because an increase in memristance induces more voltage, but this voltage cannot exceed the threshold at the same time.

The circuit in the study by Gu and Li [80] offers a short delay with extremely low energy per transition in normal mode. Unfortunately, according to the information provided, it is not comparable with our work in terms of power. Besides this, the use of a modified threshold memristor cannot guarantee that the circuit will be implementable. Moreover, this threshold is lower than the supply voltage, which increases the risk of UT problems when memristance exceeds the upper limit. More circuits may be required to prevent the UT problem resulting in higher power dissipation and larger chip area.

We propose two implementable designs based on empirically extracted memristor parameters that contain wide ranges of threshold voltage and the UT effect. Their delays are easy to tune by controlling either the pulse width or the pulse count on a pair of control signals. The memristor positions prevent short circuit paths and therefore reduce power consumption in normal mode, which is our main focus.

From the comparison table, note in particular that the dynamic power dissipation of the MemDE is one-third lower than that of the 3-bit MuxDE, which is designed based on the same technology. While our design has higher static power consumption, this amount is negligible compared to the dynamic case which is greater by 7-orders of magnitude. Furthermore, even though our MemDE is based on older technology, it uses 6-orders of magnitude less static power than the CSDE. Lastly, our MemDE occupies two times smaller area than that of 3b-MuxDE; thus, it provides a better opportunity for reducing the size of IoT devices.

To eliminate the UT problem in the above design, we have proposed a MemDE-UTA design. This consumes nearly equal static power to the 4-bit MuxDE while spending thirteen times less dynamic power, because its transistor count is about one-fifth of that of the MuxDE. Improvements in this design include the removal of UT while balancing the rise and fall delays. Interestingly, the static power consumption of the MemDE-UTA is three orders of magnitude higher than that of our first MemDE which is implemented in the larger technology. This high leakage power might be a characteristic of UMC 180nm technology, because the 4-bit Mux DE also consumes power at the same rate.

The area saving is also improved in this design as 28% of the chip area can be reduced compared to the 4b-MuxDE circuit. Therefore, our MemDE is a solution for designing area-efficient IoT devices. Our circuit size may be larger than that of the design by Gu and Li [80] due to the larger technology node. However, it can avoid the UT problem which may exist in their design. Note that the chip areas reported did not include the areas required by the memristors because the commercial technology libraries do not support these devices at the moment. Nevertheless, the memristor size may not be a critical issue because several reported memristors (up to 200μm^2 [27]) are smaller than our circuits. They can be fabricated on top of our circuits using the back-end-of-line (BEOL) process [142].

3.6.2 Excitation time model

The features of the circuits for interfacing memristors that are found in recently published papers are listed in Table 3.8. The majority of the publications rely on comparators as real-time completion detectors. Hence, they can achieve low levels of errors without the need to estimate excitation timing.

Table 3.8: Feature comparison of the tuning techniques

Work	Technique	Technology (μm)	Memristor model	Max error (%)
[123]	Comparator	N/A	Strukov *et al.* [23]	0.40
[124]	Comparator	0.35	Strukov *et al.* [23]	N/A
[125]	Comparator	N/A	Vourkas *et al.* [143]	1.00
[51]	Comparator	N/A	Joglekar *et al.* [144]	N/A
[54]	Comparator	N/A	VTEAM [34]	3.59
[52]	Pulse-based	N/A	Lehtonen *et al.* [91]	N/A
This	Pulse-based	0.35	VTEAM [34]	7.00

Regarding the pulse-based tuning approach, error data is only available for our work. The maximum error of our model is higher than that of all the comparator-based solutions. This is mostly due to the use of an average value of transistor channel resistance for the whole range of memristance. However, using pulse count for memristance programming can significantly reduce area and power consumption, because completion detection circuitry is not required.

The acceptable accuracy depends on the IoT application and the available memristance range. For instance, the memristor can be applied as multi-bit memory to reduce

the device's size and power consumption as multiple data can be stored in a memristor without any power requirements, unlike the traditional memory elements. Therefore, the higher tuning accuracy can increase the data resolution causing a smaller chip area and lower power dissipation. Recently, Stathopoulos *et al.* reported their Al_xO_y/TiO_2 memristor could store up to 6.5 bits of data within the memristance range between 20kΩ to 120kΩ [27]. In this particular case, the programming technique proposed by Berdan *et al.*, which offers the maximum error of 0.4% [123], may be employed.

The over-tuning is caused by overshoot signals between the memristor's terminals. This effect appears in the small technology because gate-drain leakage becomes significant [145]. To improve the accuracy of the excitation time model, the overshoot models reported elsewhere [146, 147] must be included.

3.7 Summary

We conceptually propose a design of MemDE that is based on the memristor and uses pulse control in tuning delay. We identify the UT problem and propose a solution in order to improve power efficiency. Based on the VTEAM model with the ferroelectric parameter set and high voltage AMS 0.35μm technology, the experiments provide circuit characteristics, including an effective delay range of 5.48ns to 13.54ns within 6 tuning steps, an average delay of 1.34ns per step, and a minimum tuning pulse width of 3ns. It is also shown that the energy used is mostly spent in the tuning mode.

We further investigate the circuit structure that is free from UT problem while retaining approximately equal rising and falling delays. Our MemDE-UTA successfully achieves this requirement. The simulation results based on UMC 180nm technology and the VTEAM model with the fitting parameters for titanium dioxide memristors reveal the circuit characteristics, including an effective delay range of 0.55ns to 1.44ns within 25 and 47 steps for tuning up and down respectively. They also show average delays for tune up and down operations of 36ps and 19ps per step respectively, and a minimum tuning pulse width of 200ps.

The excitation time model for pulse-based memristance tuning helps us to accurately determine the width of the tuning pulse. Pulse-based tuning is beneficial for the maximisation of memristor storage capability and in minimising energy and area

requirements since there is no need for memristor state verification. This chapter also contributes the GMET circuit to evaluate the proposed model, and the simulation results are close to the model's predictions. They also reveal the early-saturation issue which occurs when the voltage drop across the memristor is below the threshold due to the effect of the voltage divider. We suggest the use of high ON memristance as a solution. In addition, the proposed method works well with estimating memristance shift based on the excitation voltage. Modelling the memristance shift with respect to the overshoot signal is a subject for future work.

Our MemDE-UTA can be used to adjust the timing of digital circuits depending on the effects of parametric variations. It offers balanced rising and falling delays, low power consumption, and no unintended tuning. The use of the memristor is suitable for IoT devices that are equipped with limited or unstable power sources because the delay configuration can be retained in the memristor when power is disrupted. Furthermore, there is no need to spend power and time to re-initialise the delay when the power returns. Overall, our MemDE-UTA is an energy-efficient solution for the mitigation of the effects of parametric variations on IoT devices, which is our challenge (i).

Chapter 4

Modelling temperature effect on the memristance

This chapter describes a simple yet effective temperature model of the memristor to enable a study of temperature compensation and sensing for IoT applications. Section 4.1 analyses and integrates temperature sensitivity into the existing model. The temperature response of OFF memristance, which is dominated by titanium dioxide/chalcogenide materials, is verified based on a titanium dioxide memristor from the literature in Section 4.2. In Section 4.3, temperature experiments with silver-chalcogenide memristors are conducted to validate the model's accuracy and the relevant parameters are extracted. Finally, this chapter is summarised in Section 4.4.

4.1 Model analysis

The effect of temperature on memristance is analysed and embedded in the VTEAM model because it allows precise estimations of all reported physical device behaviour, such as linear ion drift [23, 90], nonlinear ion drift [91] and the Simmons tunnel barrier [92], while also exhibiting better computational efficiency [34]. VTEAM utilises the memristance equations (2.1) and (2.2), which are recapitulated here for convenience:

$$R_m = R_{on}s + R_{off}(1-s) \tag{4.1}$$

$$s = \frac{w(t)}{D} \tag{4.2}$$

where s is the state variable, D is the distance between the lower and higher bounds w_{on} and w_{off} of the device, R_{on} and R_{off} are the memristance values corresponding to those bounds. Furthermore, equation (2.4), which is used to estimate the shift of the doped region $w(t)$, is re-written in equation (4.3) where $v(t)$ is the applied voltage, v_{on} and v_{off} are threshold voltages, $f_{on}(w)$ and $f_{off}(w)$ are window functions and the remaining variables are fitting parameters.

$$\frac{dw(t)}{dt} = \begin{cases} k_{off}(\frac{v(t)}{v_{off}} - 1)^{\alpha_{off}} f_{off}(w) & ,0 < v_{off} < v \\ 0 & ,v_{on} < v < v_{off} \\ k_{on}(\frac{v(t)}{v_{on}} - 1)^{\alpha_{on}} f_{on}(w) & ,v < v_{on} < 0 \end{cases} \tag{4.3}$$

Initially, this work focuses on the valence change memory (VCM)-based memristor because it provides a variable resistance, which is useful for multi-level and neuromorphic computing applications [148] . In particular, titanium dioxide memristor is one of the widely used VCM devices [149]. From previous research [150], ON memristance decreases slightly with increasing temperature because the conductive filament exhibits a metallic-like conduction mechanism. Furthermore, the temperature effect on R_{on} is negligible compared to the one on R_{off} [31, 96, 151]. For this reason, the following analysis studies only the embedding of the thermal effect on R_{off} into the VTEAM model. Note that the thermal effect on R_{on} can be easily embedded in the model by following the same procedure.

The temperature analysis starts by considering the OFF memristance R_{off}, which can be expressed in the basic resistance equation as follows:

$$R_{off} = \rho \times \frac{L}{A} \tag{4.4}$$

where ρ, L and A are resistivity, length (width in this case) and cross-sectional area of the memristor respectively. The resistivity of the active layer, made from either metal-oxide or chalcogenide material, can be represented empirically by the reciprocal of the conductivity in equation (2.5) [31,35,36,38,39], as shown in equation (4.5).

$$\rho = \rho_0 e^{\frac{E_a}{k_B T}} \tag{4.5}$$

where, ρ_0 is constant, E_a is activation energy, k_B is the Boltzmann constant and T is temperature in Kelvin. In real circumstances, however, the ions may not completely migrate back to the doped region. These remaining ions reduce the value of R_{off} and therefore the effective R_{off} ($R_{off,eff}$) can be expressed as:

$$R_{off,eff} = R_{off} - R_c \tag{4.6}$$

where R_c is a constant representing the offset caused by the remaining ions. Substituting equations (4.4) and (4.5) into equation (4.6) yields:

$$R_{off,eff} = (\rho_0 e^{\frac{E_a}{k_B T}} \times \frac{L}{A}) - R_c \tag{4.7}$$

This equation shows the temperature-based volatility where the memristance declines exponentially with increasing temperature, which agrees with the property reported elsewhere [152]. Also, it can be applied to other materials such as tantalum [151] and manganite [153]. As the impact of temperature on R_{on} is negligible, equation (4.7) is used to represent the effect of temperature on R_{off} in equation (4.1). Hence, the memristance model is modified as follows:

$$R_m = R_{on}s + ((\rho_0 e^{\frac{E_a}{k_B T}} \times \frac{L}{A}) - R_c)(1 - s) \tag{4.8}$$

This is the final model which describes memristance as a function of temperature. Notably, this chapter only uses $s = 0$ because we focus only on R_{off}. As a result, R_m is equal to $R_{off,eff}$. Note that Li *et al* [33] reported R_{off} of titanium dioxide memristor is independent of the cross-sectional area due to the random distribution of oxygen vacancies in the undoped region [94,95]. Discarding A does not affect our model because ρ_0 can absorb its value. To be general, however, we will keep this parameter untouched.

4.2 Model verification based on titanium dioxide memristor

To verify the model, firstly, equation (4.4) is used to calculate the resistivity of titanium dioxide (16,000Ωm) using published data [33] (R_{off}=4MΩ, A=10×10μm^2 and L=25nm). Then, values of R_{off} at different device widths are estimated using the calculated

resistivity compared to the experimental data [96] as listed in Table 4.1. From the table, the maximum error of 1.25% is at L=20nm, which demonstrates the accuracy of the proposed analysis.

Table 4.1: Calculated OFF resistance of titanium dioxide memristor

Width L (nm)	R_{off} from [96] (MΩ)	Calculated R_{off} (MΩ)	Error (%)
10	2	1.98	1.00
20	4	3.95	1.25
30	6	5.93	1.10
40	8	7.90	1.20

Table 4.2: Thermal effect on resistivity of titanium dioxide memristor

Temperature (K)	R_{off}/R_{on} ratio from [96]	Calculated R_{off} (kΩ)	Calculated resistivity (Ωm)
298	15.0×10^3	1,950.00	158.00×10^2
338	4.0×10^3	520.00	42.10×10^2
358	1.0×10^3	105.00	10.50×10^2
398	0.5×10^3	0.53	5.27×10^2

Table 4.3: Calculated values for the constants in equation (4.5)

ρ_0 (Ωm)	E_a (meV)	R^2(%)
0.0122	360	96.00

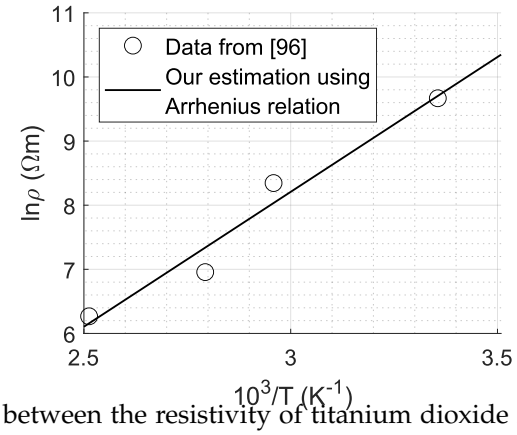


Figure 4.1: Comparison between the resistivity of titanium dioxide memristor in Table 4.2 and Arrhenius relation in equation (4.5) using the extracted parameters in Table 4.3. The resistivity in Table 4.2 is calculated from the R_{off}/R_{on} ratios estimated by Abunahla *et al.* [96]. These values are validated using the data measured from the fabricated device in [151].

Secondly, values of R_{off} and resistivity at different temperatures are estimated using the extracted values (R_{on}=130Ω, A=9×9μm^2, L=10nm and R_{off}/R_{on} ratios) [96] as listed in Table 4.2. Finally, the parameters ρ_0 and E_a in equation (4.5) are calculated numerically using the resistivity values obtained and the non-linear least square fitting tool in

Table 4.4: VTEAM model parameters (partially from [109])

Parameter	Value	Unit
α_{off}	4	–
α_{on}	4	–
v_{off}	0.3	V
v_{on}	-1.5	V
k_{off}	0.091	m/s
k_{on}	-216.2	m/s
$w_{off} = L$	3	nm
w_{on}	0	nm
R_{on}	1	kΩ
A	12×12	μm^2

MATLAB. Both parameters determined for titanium dioxide are tabulated in Table 4.3, with an accuracy R-squared of 96%. Both values agree with those in the report by Mardare *et al.* [154]. The graphs in Figure 4.1 show that the resistivity in Table 4.2 and the model based on equation (4.5) using the parameters in Table 4.3 are consistent. Although the obtained parameters fit the data from earlier work [96], the comparison of accuracy with other research cannot be accomplished because the publications concerned do not provide sufficient information. For example, one study [152] reports only the trend of temperature and resistivity, but does not provide exact values. By using the same procedure defined here, the parameters of equation (4.5) can be recalculated for memristors with different materials such as HfO_2 [31], ZnO [96] and Ta_2O_5 [96].

Embedding the modified model in equation (4.8) with the VTEAM model in equation (4.3) enables a thermal analysis using a circuit simulator and computing software such as Cadence Spectre and MATLAB. The Verilog-A code of the model is provided in the Appendix.

A 100MHz sine wave 2.5V in amplitude is applied as an input of the memristor to simulate the I-V relationships at 298K, 338K, 358K, 398K (25°C, 65°C, 85°C and 125°C) using Cadence Spectre. The VTEAM parameters are retrieved from a previous study [109] and are summarised in Table 4.4. They fit the physical devices subsequently reported [24]. In addition, both L and w_{off} are the same parameters, which is the device width. The cross-sectional area A is selected such that R_{off} (343kΩ) becomes close to the original parameter (300kΩ). The I-V simulation results in Fig. 4.2 show that memristance decreases at high temperature, resulting in the wider positive (right) lobes and the steeper R_{off} slopes in the negative (left) lobes.

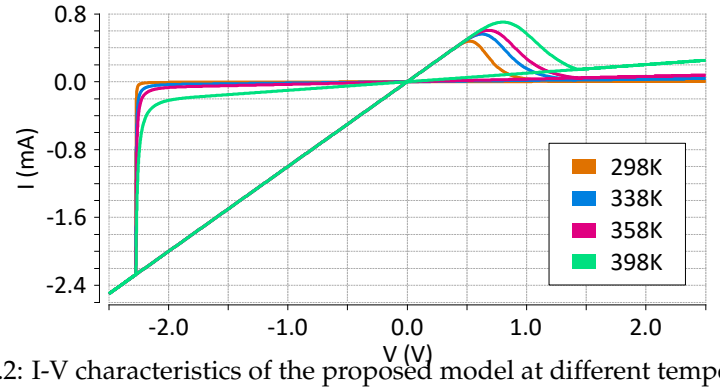
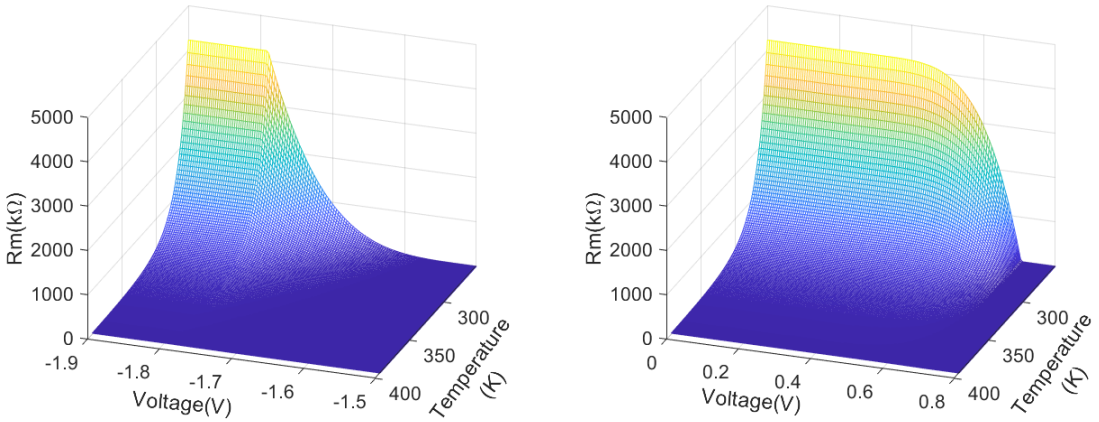


Figure 4.2: I-V characteristics of the proposed model at different temperatures.



(a) (b)

Figure 4.3: Relationship between temperature, applied voltage and memristance with an excitation time of 10ns: (a) tune memristance up; (b) tune memristance down.

The thermal analysis using MATLAB is illustrated in Fig. 4.3 showing the memristance tuning runs for up and down directions with an excitation time of 10ns. Both sub-figures show that memristance declines exponentially when temperature rises. Furthermore, saturation is achieved when the temperature reaches approximately 358K ($R_m \approx$32kΩ). Moreover, the memristance changes non-linearly depending on the voltage applied, as described in the original VTEAM model.

In summary, the VTEAM model is modified to include the effect of temperature based on data in the literature. It shows that temperature impacts the resistivity of the material and subsequently the memristance also. Therefore, temperature compensation is needed when using the memristors at temperatures different from room temperature. Our model enables such compensation, which significantly improves circuit accuracy.

4.3 Temperature experiments on silver-chalcogenide memristor

In this section, silver-chalcogenide memristors are used in our temperature experiments to extracts the model parameters based on equation (4.7). There are three experiments in this section: high-temperature sweep, low-temperature sweep and constant temperature. Firstly, the high-temperature experiment is conducted to observe the change in memristance with temperature and also to inspect the memristance hysteresis between rising and falling temperatures. Then, the low-temperature experiment extends the model's coverage. Both experiments were repeated four times to ensure the reliability of the results, and the data is combined to extract the model parameters. Finally, values of memristances at different temperatures over time are measured to observe the consistency.

4.3.1 Experiment setup

For the experiments, we employed two tungsten-based (*W*) silver-chalcogenide memristor chips in ceramic packages from Knowm Inc. Two out of eight memristors from each chip were selected: memristor#4 and #8 from chip#2 (C2M4 and C2M8), and memristor#1 and #5 from chip#3 (C3M1 and C3M5). Before the experiments, the functionality of the memristors was tested using the official toolset which consists of Analog discovery 2 and the Memristor discovery board and its software as shown in the left-hand side of Fig. 4.4. Note that the memristor chip is attached to the ZIF socket at the centre of the board. Each memristor receives a series of erase, write and erase pulses which are 50ms-width half-sine waveforms. The write and erase amplitudes are 1V and -2V respectively. Each step is followed by a 100mV square wave to read the memristance. Samples of the test results are depicted in the right-hand side of Fig. 4.4. They show that C2M4 and C3M1 passed the test as their high memristances (infinity) are read after erasing while the low memristances in tens of kilohms are detected in the writing step.

The experimental set-up in Fig. 4.5 illustrates the overall set-up. A Keithley SourceMeter 2612B was employed to measure the memristance in every experiment. It is programmed through its web interface to read memristance using 10mV signal. A digital thermometer reads the temperature next to the memristor chip during memristance measurements. The memristor chip, temperature probes and test probes of

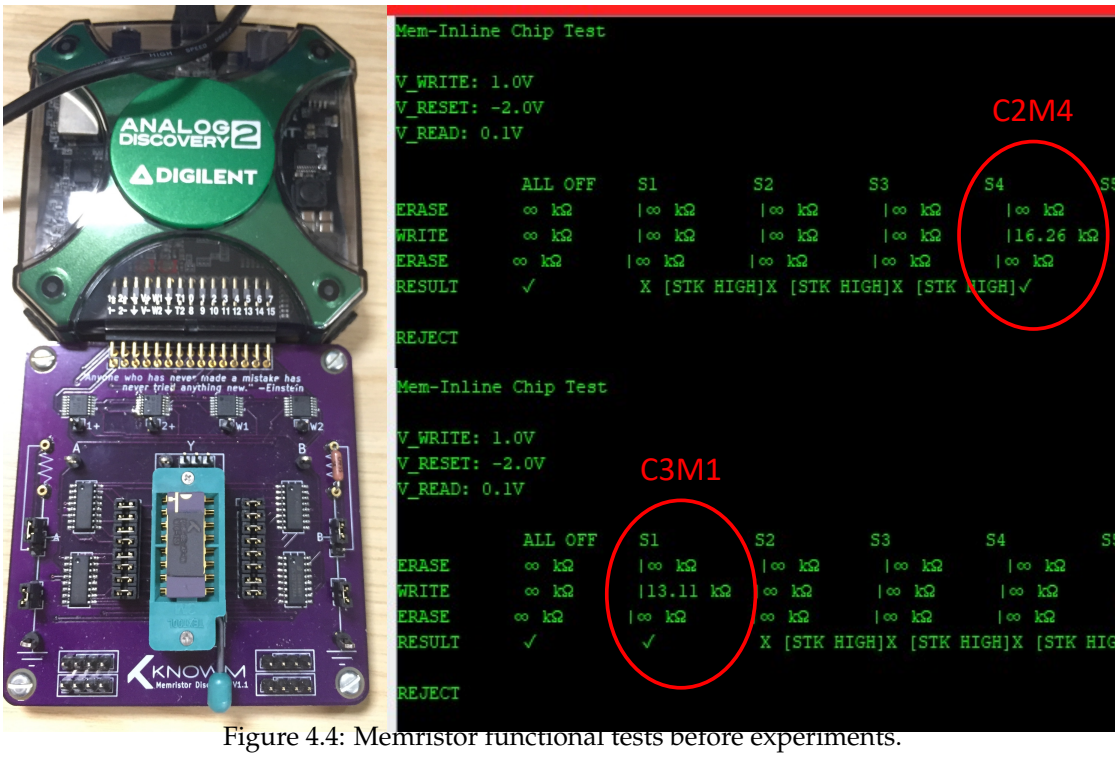


Figure 4.4: Memristor functional tests before experiments.

Table 4.5: Temperature sweep experimental set-up

Parameter	Value			
Wafer	W20			
Chip number	2		3	
Memristor number	4	8	1	5
Measuring points (K)	Low: 253, 278 High: 298, 318, 338, 353, 378			
Temperature cycle	3			
Sampling frequency (Hz)	33			
Samples per reading	5			
Reading voltage (mV)	10			
Setup time for read signal (ms)	Low: 450 / High: 30			
Temperature settling time (mins)	Low: 60 / High: 15			
Repeat experiment	4			

the Keithley SourceMeter 2612B are placed inside the oven, as shown in Fig. 4.6. The chip is attached to a PCB that is connected to the test probes. The main temperature sensor probe is placed close to the chip and another probe is attached to the aluminium plate at the bottom for the reading of the heat source. In the high-temperature experiment, the chip is covered by an aluminium foil to shield the memristor from noise and to stabilise the temperature around the chip (the plate is to be placed in the oven). In the low-temperature experiment, a plastic box is used instead of the aluminium foil and

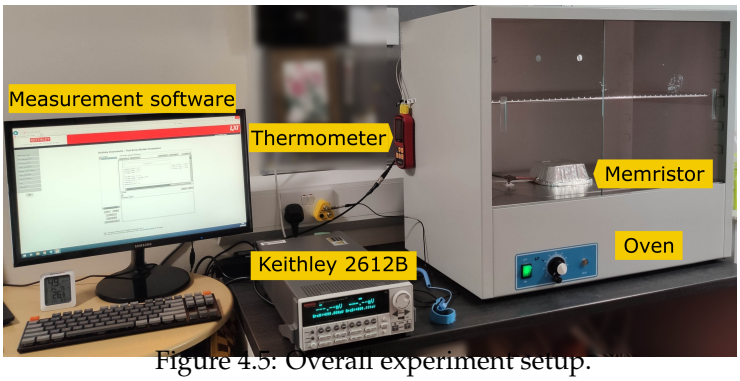


Figure 4.5: Overall experiment setup.

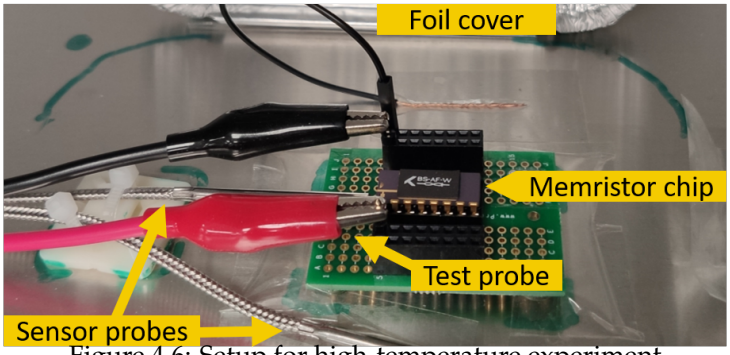
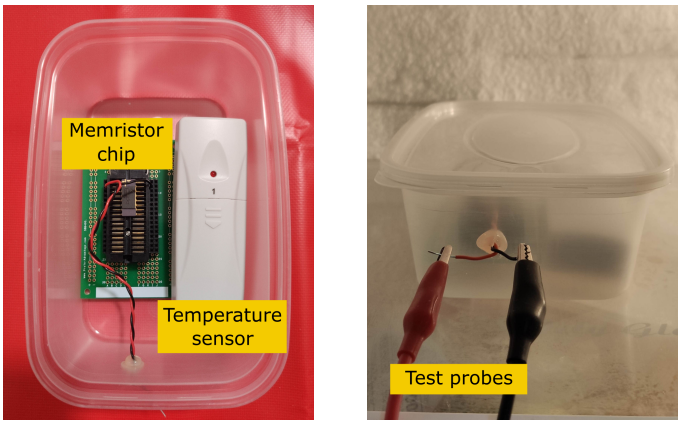


Figure 4.6: Setup for high-temperature experiment.



(a) (b)

Figure 4.7: Setup for low-temperature experiment: (a) inside the plastic box; (b) measurement in action.

plate to protect the memristor from moisture (the box is to be placed in a fridge-freezer) as depicted in Fig. 4.7a and 4.7b. Note that the wireless temperature sensor in Fig. 4.7a is applied instead of the basic one in Fig. 4.5 in order to minimise the number of holes, which cause temperature leakage. The temperature read from the wireless thermometer is 0.5°C higher than the basic one. In addition, our measured memristances might be affected by the moisture, which inversely relates to the temperature.

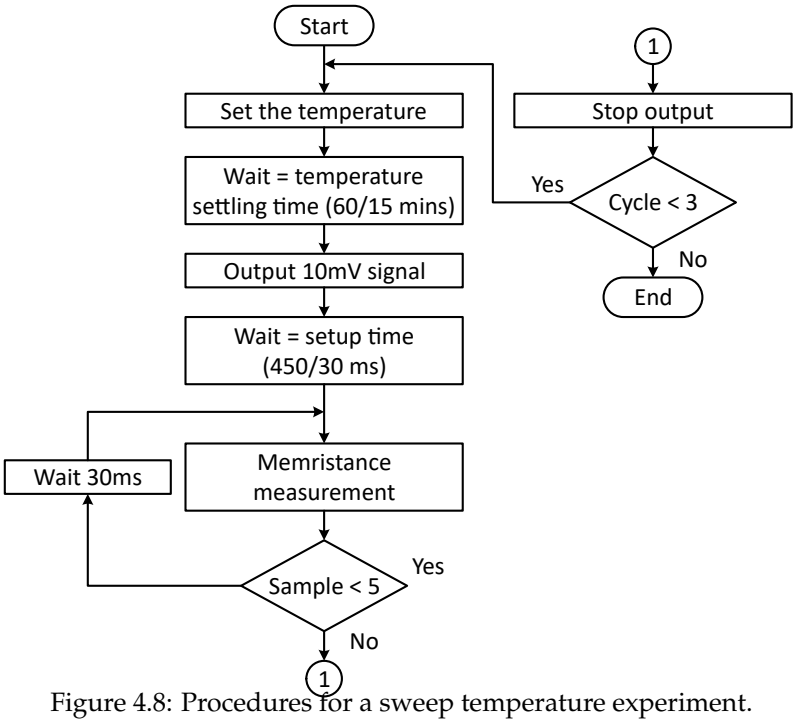
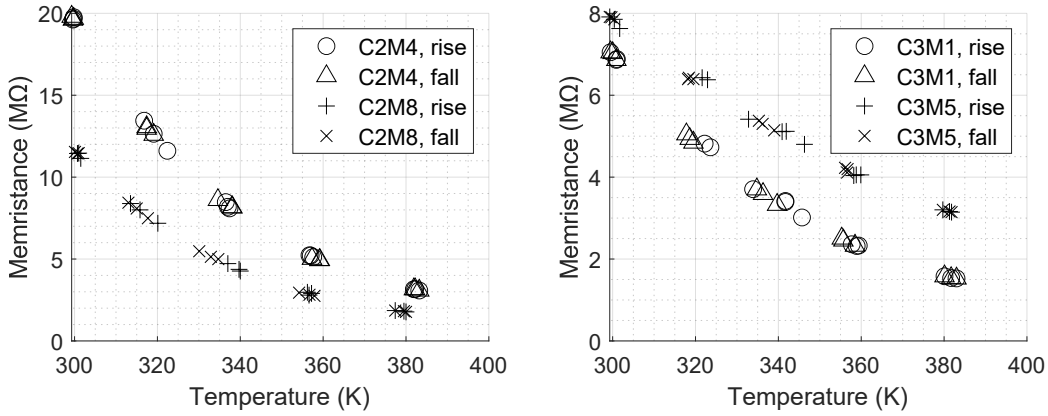


Figure 4.8: Procedures for a sweep temperature experiment.

The procedures for both high- and low-temperature sweeps follows the flowchart in Fig. 4.8. The temperature is swept from between low and high values for three cycles and a low reading voltage of 10mV is applied at each measuring point. This voltage is chosen to provide sufficient margin below the memristor threshold, which decreases with temperature to >80mV at 423K [155]. Furthermore, our reading test with regular resistors in the MΩ range has shown that our equipment can read the resistances correctly using the specified reading voltage. After adjusting the temperature (high-temperature experiment), the oven is left untouched for 15 minutes to ensure the chip's temperature is stable. This is confirmed by monitoring and measuring the chip's temperature before and during the memristance reading. The same procedures are done for the cold-temperature experiment, except the memristors are left in the refrigerator and freezer for 60 minutes. Then, the memristance, which is a proportion of voltage and current, is sampled five times with a sampling period of 30ms and the obtained values are averaged to minimise the noise effect. Note that memristance is extremely high at low temperature (tens of MΩ); therefore, the required set-up time in the low-temperature experiment (450ms) is longer than in the high-temperature one (30ms). Furthermore, a settling time of the minimum temperature between measuring points is required in order



(a) (b)

Figure 4.9: High-temperature sweep results for: (a) C2M4 and C2M8; (b) C3M1 and C3M5.

for the temperature to stabilise. The experimental set-up for the temperature sweep is summarised in Table 4.5.

In the fixed temperature experiments, each memristor is placed in a constant temperature of approximately 323K, 338K, 353K or 378K to inspect the consistency of memristance at different temperatures. Memristance is read every 15 minutes for at least 6 hours (24 readings). The sampling frequency and voltage are the same as those for the temperature sweep experiment.

4.3.2 Experimental results and discussions

Hysteresis

To determine the hysteresis, data from the high-temperature sweep is plotted separately between rising and falling temperatures, as illustrated in Fig. 4.9. The subfigures reveal that the memristances in both directions are inline; hence, it can be concluded that hysteresis regarding the temperature of Ge-rich material (Ge_2Se_3 in this experiment) is negligible. In contrast, such hysteresis is observed in Se-rich material ($Ge_{22}Se_{78}$ in [156]), which is also used as active layers of memristive devices, e.g. conductive bridge random access memory (CBRAM) [105]. Note that in order to maintain point visibility, the data in both plots is from the first experiment only; the other experiments give similar results.

Full temperature sweep and parameter extraction

Combining high- and low-temperature sweep data creates a full sweep as depicted in Fig. 4.10. Note that the memristances are shown in logarithmic scale. All subfigures confirm the exponential relationship between memristance and temperature, as expected based on equation (4.7). To extract the model parameters, ρ_0, L and A in equation (4.7) are lumped as β for simplicity. The parameters for each memristor are extracted using non-linear least square fitting in MATLAB. The extracted parameters are listed in Table 4.6 and the models are plotted as solid lines in Fig. 4.10. It can be seen that the model fits the data very well, with the minimum R-squared value of 88.64%. Finally, ρ_0 is calculated based on L = 300Å and the via size of 3µm [41] as listed in the table. Note that the extracted activation energies are higher than the value reported in [155], which is 31meV. The reason for this is subject to future work.

Table 4.6: Extracted temperature model parameters

Memristor	β	ρ_0 (Ωm)	E_a (meV)	$R_c(\Omega)$	R^2 (%)
C2M4	6.88×10^4	16.21	148.74	3.52×10^6	88.64
C2M8	4.85×10^3	1.14	205.61	2.96×10^5	88.95
C3M1	8.62×10^5	203.21	70.08	5.88×10^6	99.75
C3M5	4.57×10^6	1,076.98	36.82	1.09×10^7	99.61

Interestingly, the model agrees with the experimental data, but there are small disagreements with the data of C2M4 and C3M1 at 378K. This might be due to the remaining ions in the active layer. Their effects in tens of the kilo-ohms range are at least two orders of magnitude lower than that of the active layer at room temperature [157]. However, they become a concern at high temperature as the memristance drops significantly. Accounting for this effect so as to obtain more accurate models is a matter for future research. The device-to-device variation in Fig. 4.11 shows an insignificant variation in R_{off} of each memristor between before and after the temperature sweep.

Fixed temperature experiment

Because the characteristics of all memristors are the same when the temperature is below 378K, only results for C3M5 at 338K and 353K are illustrated in Figs. 4.12a and 4.12b respectively, for brevity. All plots can be found in Appendix B. The results for C2M4, C2M8, C3M1 and C3M5 at 378K are displayed in Fig. 4.12c- 4.12f. Note that, for

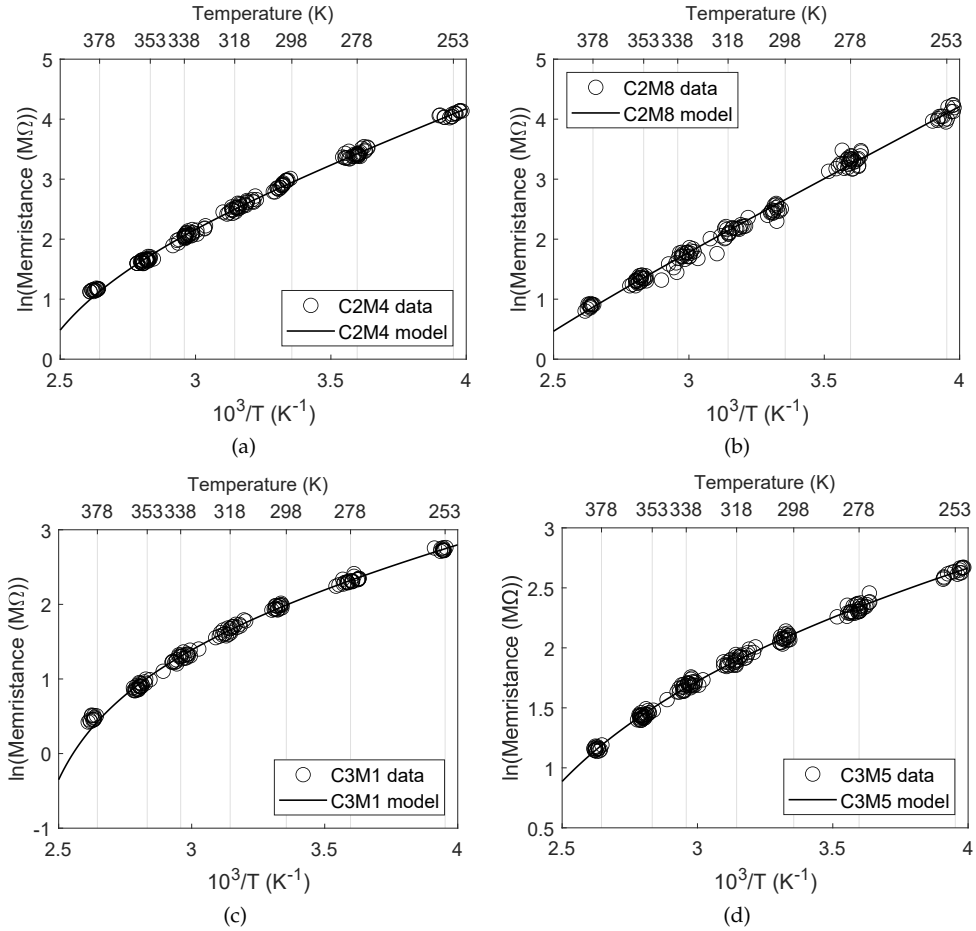


Figure 4.10: Logarithmic scale memristances between 253K and 378K, and models of: (a) C2M4; (b) C2M8; (c) C3M1; and (d) C3M5.

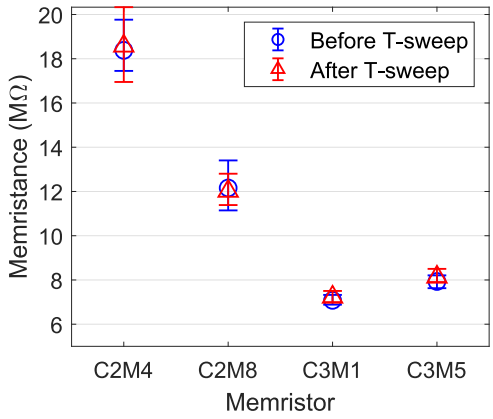


Figure 4.11: Device to device variation before and after the temperature sweeps.

visualisation purposes, memristor conductance ($G_m = 1/R_m$) is used as it changes in the same direction as temperature. In addition, the actual temperature readings are plotted

alongside each conductance diagram to show the variation during the experiments due to fluctuations in ambient temperature.

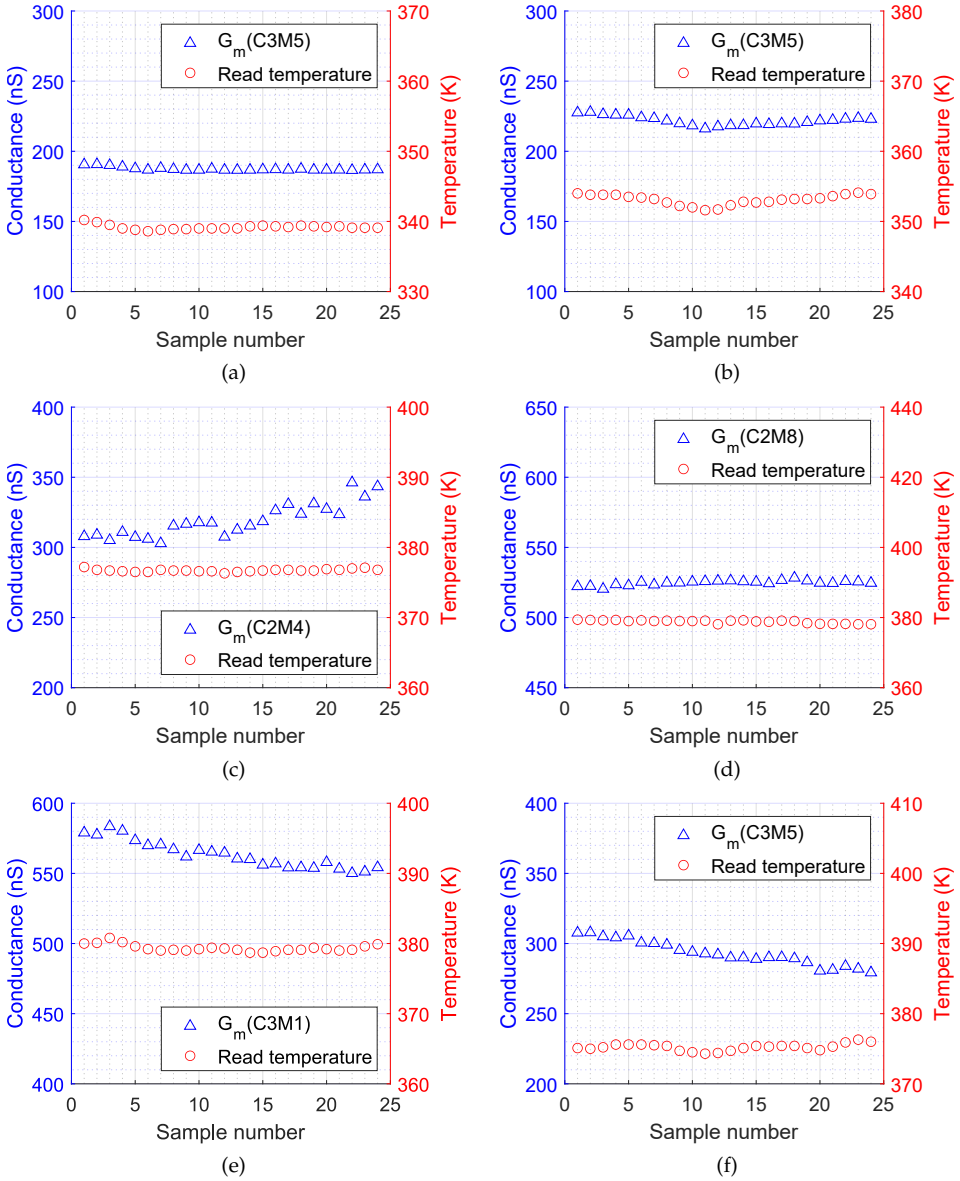


Figure 4.12: Memristor conductance and read temperature vs sample: (a) C3M5 at 338K; (b) C3M5 at 353K; (c) C2M4 at 378K; (d) C2M8 at 378K; (e) C3M1 at 378K; (f) C3M5 at 378K.

From Figs. 4.12a and 4.12b, the memristance values vary with the same pattern as the actual temperatures, and this is similar to the results from C2M4, C2M8 and C3M1. Therefore, it is concluded that memristance is stable and correlates with temperature up to 353K. Similar experiments on titanium dioxide memristor (Al_xO_y/TiO_2) were done by Stathopoulos *et al.* [27]. The results show the memristances are also stable at 358K.

Interestingly, the conductances behave differently at 378K. The conductance from C2M4 increases (Fig. 4.12c) while the one from C2M8 fluctuates over time; see Fig. 4.12d. Furthermore, decays are observed from both memristors from chip#3; see Figs. 4.12e and 4.12f. The reason for this behaviour needs to be investigated in the future. Although the stability issue exists at high temperature in the long term, it is not clear if this is the case in the short term. Indeed, the memristance levels in Fig. 4.10 are approximately the same even when temperature is cycled multiple times.

4.4 Summary

This chapter proposes a realisation of a simple thermal model of a titanium dioxide memristor based on the recently proposed physical model and experimental data. The evaluation of the model yields a maximum error of 1.25% and the extracted model parameters yield an accuracy R-squared value of 96%. Our model shows that the OFF memristance declines exponentially and becomes saturated once the temperature reaches to 358K ($R_m \approx$32kΩ). This temperature dependency is stronger than that of the transistor because the temperature-memristance relation is exponential. In contrast, the relation between temperature and transistor's threshold voltage affecting the channel resistance is quadratic. The model implementation in Verilog-A can be found in the Appendix.

This chapter also presents a validation of our model for empirical data with silver-chalcogenide memristors. Four memristors from two chips were selected for our experiments. The temperature sweep between 253K and 383K reveals that memristance changes exponentially with temperature and this relationship is similar to that of the titanium dioxide memristor in the literature. The model parameters were extracted with a minimum R-squared value of $>$ 88%. In the fixed temperature experiment, the memristances were stable up to 353K. The reason for the stability issue at 378K and improvements in accuracy for high temperature applications will be investigated in the future.

Our model accurately estimates the effect of temperature on the memristance of memristors which are built from metal-oxide or chalcogenide material. This model addresses challenge (ii), as it enables advances in memristive temperature sensing and

temperature-tolerant memristive systems. It is used in the next chapter to demonstrate the temperature effect in decoding values of memristance.

Chapter 5

Thermally-aware memristance decoder

This chapter presents a design for an energy-efficient and resolution-scalable memristance decoder. It features the temperature model in the previous chapter which is able to mitigate the decoding error caused by fluctuations in temperature. This decoder is essential in improving the energy efficiency and temperature tolerance of memristive IoT devices.

Section 5.1 introduces the concept of the decoder and its interface with the existing memristive circuits. The detailed operation and implementation is explained in Section 5.2. In Section 5.3, the data range assignment is then demonstrated and discussed. The metastability problem and its solution are tackled in Section 5.4, and in Section 5.5 the proposed decoder is simulated and its characteristics such as offset, latency and energy obtained. This section also demonstrates the offset calibration using capacitor arrays and considers the decoding issue regarding the effect of temperature. Section 5.6 compares the decoder's performance to findings in the literature and, finally, this chapter is summarised in Section 5.7.

5.1 Overview

Our decoder targets memristive systems whose outputs are in the form of currents, with an example shown in Fig. 5.1. The figure illustrates a dot-product engine based on a memristive crossbar, which is a common structure in many publications related to machine learning [1, 2, 43, 158, 159]. From the figure, each memristor cell is programmed

in the form of conductance (G_{ij}), where i and j represent its row and column respectively. To perform the dot-product operation, the input signals (V_i) in the form of voltage are fed into the cells that are in the same row. Then, the resulting currents ($I_{ij} = V_i \times G_{ij}$) of the cells in the same column are combined as I_j, which represents the output of this operation. Note that this structure is also used for multi-bit memory applications where a cell can be read by sending a reading voltage to the target row and measuring the current at the corresponding column [54, 160]. In real circumstances, the temperature will affect the memristances as described in the previous chapter. Consequently, each memristor's conductance, which is an inverse of the memristance, will be changed as well, resulting in the deviation of the output current of each memristor cell. Therefore, to read these currents precisely, our decoders need to support temperature compensation. Moreover, they need to adjust their power consumption depending on the available power to maintain the reliability of the IoT devices.

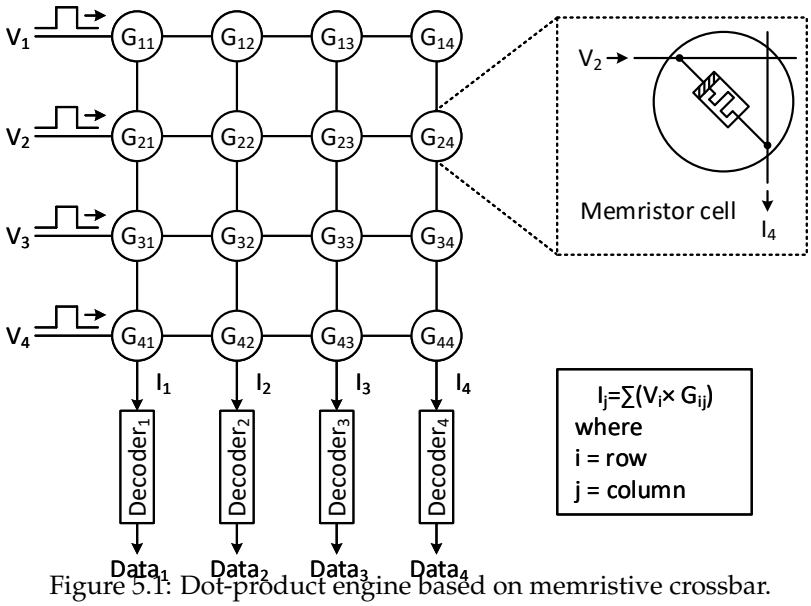


Figure 5.1: Dot-product engine based on memristive crossbar.

To decode the output current, many works convert this current to a voltage by using a series resistor [51, 54] or a trans-impedance amplifier (TIA) [161] and then feed this voltage to an analog-to-digital converter (ADC). However, this approach is not suitable for IoT applications because it requires on-chip resistors, which are vulnerable to parametric variations and occupy a large chip area. Therefore, our decoder employs a current-mode comparator to receive the output current directly. This comparator also inherits its intrinsic advantages over its voltage counterpart, such as low power,

wide bandwidth and less susceptibility to power supply fluctuations [135, 136]. The comparator is designed to provide resolution scaling based on the available power budget, and it supports both synchronous and asynchronous schemes. In addition, this design is resistor-free to save on chip area, which is a critical issue in edge device design [2], and to avoid any resistor-induced variations.

A simplified block diagram of the proposed decoder is shown in Fig. 5.2. The controller operates the comparator through *Control* signals to iteratively compare the memristor current to the reference ones adjusted according to *Data* signals. Note that the reference sources are inside the comparator block. The decoded data is then compensated for in terms of temperature variation in order to yield the correct values. The temperature compensation process is accomplished using the temperature model described in the previous chapter. Furthermore, the comparator offset which arises due to process variations is also calibrated by the controller. Note that the comparator is called the "resistance comparator" from this point onwards, as it is designed specifically to compare resistances.

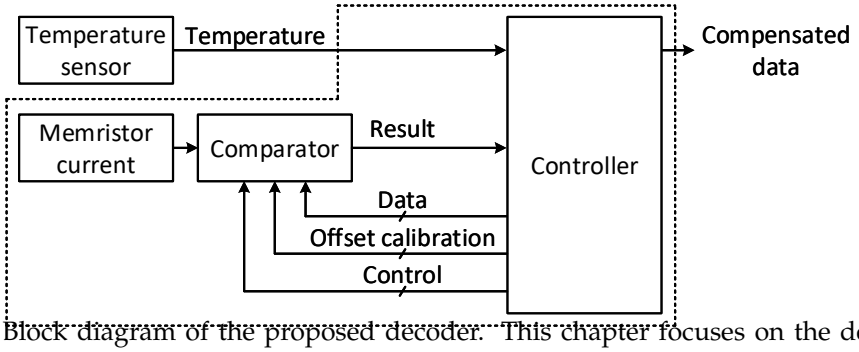


Figure 5.2: Block diagram of the proposed decoder. This chapter focuses on the design of the comparator and controller (these components are circumscribed by the dashed line).

5.2 Decoder circuit design

The block diagram in Fig. 5.2 is expanded with more detail in Fig. 5.3. The decoder consists of a controller and a resistance comparator. Similar to the successive approximation register (SAR) ADC [162, 163], the controller is used to run the search algorithm (e.g., binary search), offset calibration and temperature compensation. It uses *En* and *Cmp* to control the comparator operation which is described by the state diagram in Fig. 5.4 and Table 5.1. It also feeds the decoded data back to the digitally controlled current source (DCCS) to adjust the reference current (I_f) based on the implemented search algorithm.

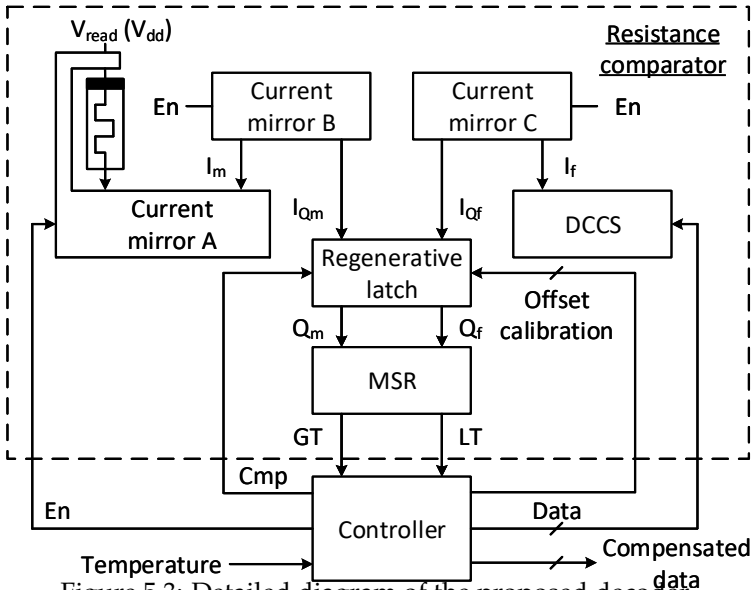


Figure 5.3: Detailed diagram of the proposed decoder.

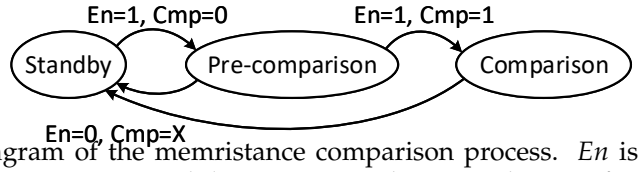


Figure 5.4: State diagram of the memristance comparison process. En is used to switch from standby to pre-comparison state while Cmp is used to switch state from pre-comparison to comparison. The state returns to standby and is ready for the next iteration once En is 0.

Table 5.1: Operation modes

Signal		Mode
En	Cmp	
0	X	Standby
1	0	Pre-comp.
1	1	Comparison

Table 5.2: Resistance comparator results

Result	Signal	
	Q_m, Q_f	GT, LT
Standby/pre-comp.	0, 0	0, 0
$I_m > I_f$ ($R_m < R_f$)	1, 0	1, 0
$I_m < I_f$ ($R_m > R_f$)	0, 1	0, 1

The resistance comparator is similar to previous designs [132, 164]. Inside, there are three current mirrors (A, B, and C) and a DCCS. The memristor is connected to the current mirror A to produce the memristor current (I_m). The current mirrors B and C convey I_m and I_f to the regenerative latch (I_{Qm}, I_{Qf}). Then, the latch compares the two currents and indicates the larger/smaller currents at Q_m and Q_f. The metastability resolver (MSR) is attached to the outputs of both latches to filter out the metastable state [134].

According to the state diagram, flowchart and circuit implementation in Figures 5.4-5.7, the decoder is initialised by calibrating the offset. Then, it enters standby mode

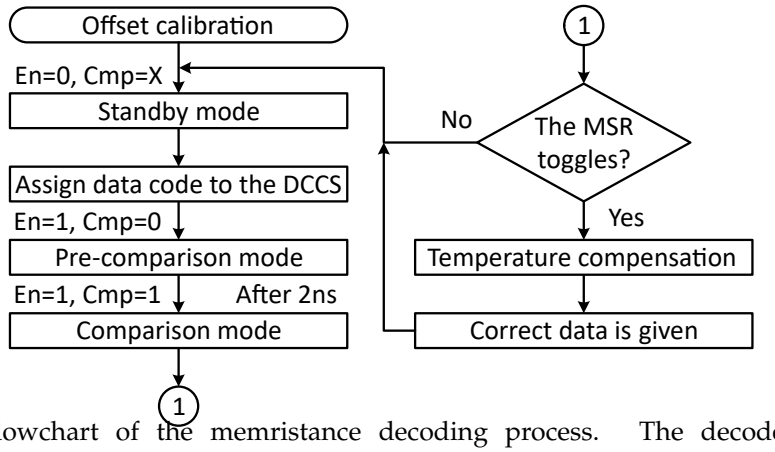


Figure 5.5: Flowchart of the memristance decoding process. The decoder repeats the memristance comparison until the latch outputs change (the uncompensated memristance is found). Temperature compensation then takes place to yield the correct value.

where En is set at 0 by the controller. This discharges the result nodes Q_m and Q_f via transistors N2-3 and disables the DCCS and all current mirrors to minimise power consumption. Then, the decoded data is initialised to set up the reference current (DCCS). Subsequently, the comparator enters pre-comparison mode by toggling En to 1 so as to enable all current mirrors and the DCCS. As a result, I_{Qm} and I_{Qf} flow through the latch towards the ground. In order to ensure the stability of both currents, this state has to be sustained for a certain amount of time (2ns).

The comparison starts by switching Cmp to 1 to disable N2-3. As a result, I_{Qm} and I_{Qf} flow against each other at Q_m and Q_f. At this moment, the metastability, where the voltages at Q_m and Q_f change to a non-Boolean level between V_{dd} and ground, occurs and may cause an incorrect interpretation in the successive building blocks. The latch's outputs indicating the larger/smaller currents are valid once this metastable state is resolved. To filter the metastability, the MSR in Fig. 5.8 is applied, and its outputs are given as GT and LT, as summarised in Table 5.2. The controller configures the data, which feeds back to control the DCCS. Then, it repeats the comparison process until the outputs (GT and LT) are different from the previous ones. This indicates that the data has been found. Finally, it adjusts the decoded data regarding the sensed temperature, which is modelled in the previous chapter, to yield the correct value. Table 5.2 summarises the relationship between the memristor and reference currents (I_m, I_f), the latch's outputs (Q_m, Q_f), and the MSR's output (GT, LT).

To support resolution scalability, the reference current (I_f) is configurable, as it sums

the currents from the base and the DCCS (Fig. 5.6). The DCCS is implemented as multiple banks of current sources. Each bank contains a number of parallel current sources (Fig. 5.7) with respect to the number of bits denoted by m. Also, each bank is enabled by the heading transistor which is controlled by Enm signals. The current sources are programmed by the decoded data ($Data$) from the controller. This design allows the controller to select the resolution according to the available power.

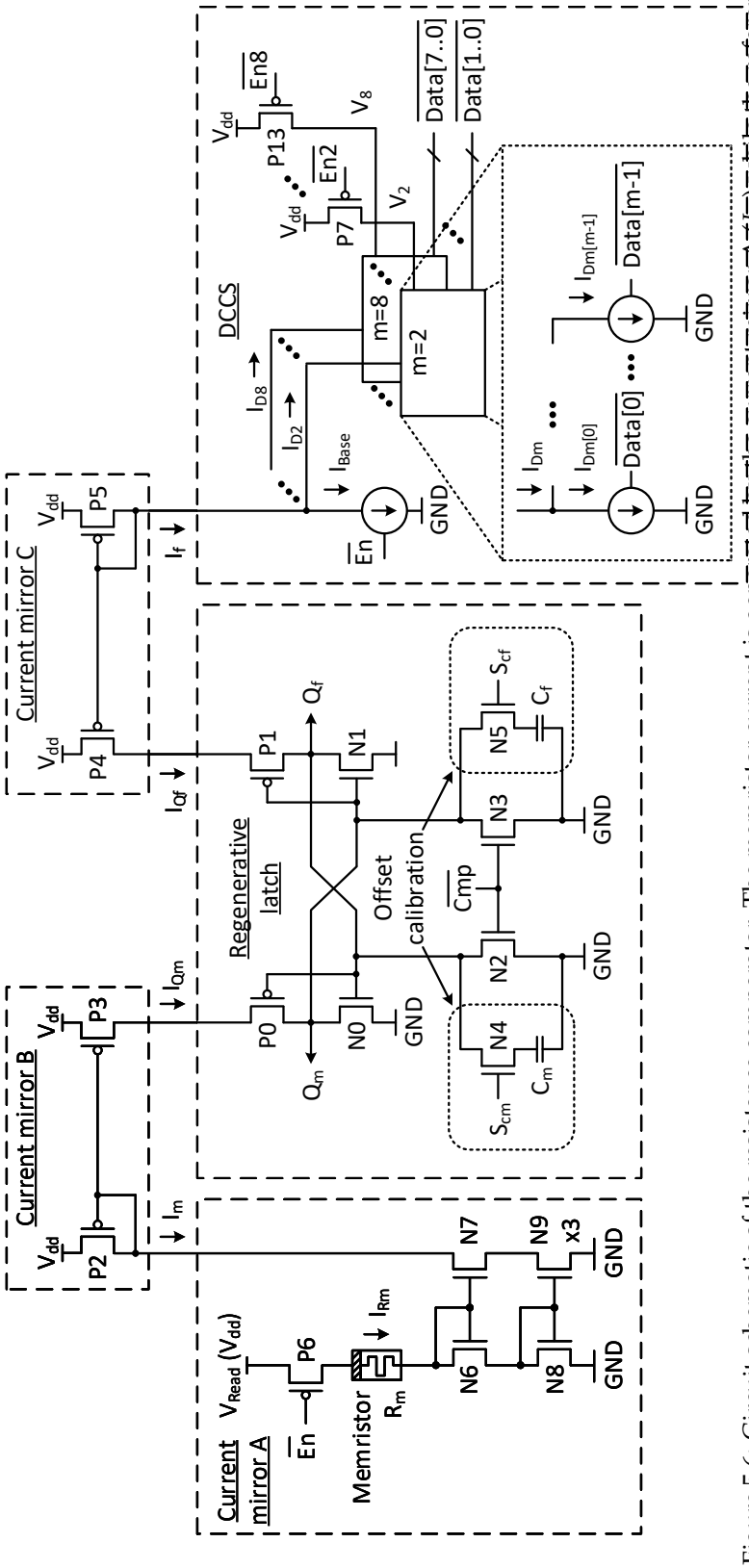
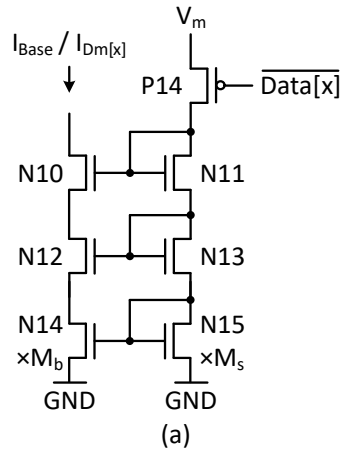


Figure 5.6: Circuit schematic of the resistance comparator. The memristor current is generated by the current mirror A (I_m) while the reference current is produced by the DCCS (I_f). The DCCS contains multiple banks of current sources to support multiple resolutions which can be selected by Enm (En in Fig. 5.3- 5.5). Note that m denotes the number of current sources in each bank; I_m and I_f are copied to I_{Qm} and I_{Qf} which are compared by the regenerative latch.



Bit	Mb	Ms
Base	12	1
Data[0]	6	4
Data[1]	6	1

(b)

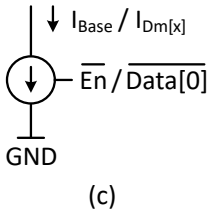


Figure 5.7: Implementation of the current sources in DCCS: (a) circuit schematic, where $I_{Dm[x]}$ is controlled by the numbers of N14 and N15 (M_b and M_s); (b) numbers of transistors for the base and data bits; (c) symbol of the current source.

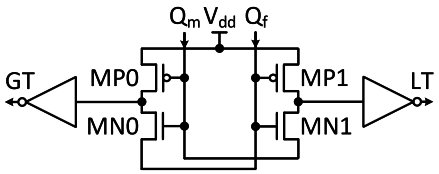


Figure 5.8: Schematic of the metastability resolver (MSR). MP0 and MP1 are ON and V_{dd} is connected to both inverters even when both inputs (Q_m, Q_f) enter the metastable state. Once the metastability vanishes, the complementary outputs are sent to both inverters.

Regarding circuit implementation, the transistors P2/P5 and P3/P4 have to be matched to avoid any current offsets. To minimise the offset caused by mismatch between the current mirror A and the DCCS, both sub-circuits are isolated from the latch by the current mirrors B and C. The width and length of all transistors in our implementation (Fig. 5.6-Fig. 5.8) are 80nm and 60nm respectively.

5.3 Data range and decoding

The decoder is designed for a specific range of memristance. It provides different levels of precision based on the energy available and the accuracy required. The DCCS consists of different digitally controlled current mirrors, as mentioned in the previous section. Resolution is calculated using the relationship: $NumberOfBits = log_2(MemristanceRange/StepSize)$. Then, the current generated by each data boundary is observed and used to determine the transistor size of each current source (Fig. 5.7). An example of 2-bit code for a memristance range of 134.257kΩ - 324.801kΩ with a step size

of approximately 80kΩ is shown in Table 5.3. A safety gap must be allocated at each end to avoid the uncertainty of R_{on} and R_{off} due to process variation. Furthermore, the gap at the low memristance side should be widened to ensures that saturation (where low memristance causes insufficient voltage for memristor programming) does not occur, as discussed in Section 3.5.3. Note that the memristance in Table 5.3 defines the upper bound of the range. For example, data 00 is defined between 237.506kΩ-324.801kΩ. In addition, the upper bound of 11 is higher than the specification (80kΩ gap) because the combination of $Data[0]$ and $Data[1]$ yields a higher current.

Table 5.3: Data range, latency and energy at 298K (25°C)

Data [1..0]	Expected R_m ($k\Omega$)	Effective R_m ($k\Omega$)	State variable	Latency (ns)		Energy (fJ)	
				Worst case	Best case	Worst case	Best case
00	324.801	326.171	0.95	2.91	2.36	30.73	26.53
01	237.506	234.929	0.69	2.90	2.30	39.00	32.93
10	167.761	164.908	0.49	2.77	2.27	44.93	38.46
11	134.257	131.691	0.39	2.60	2.19	48.79	46.23

5.4 Metastability

The proposed comparator enters the metastable state during decision making, as revealed in Fig. 5.9 (Q_m and Q_f). Its magnitude is greater and its duration longer when the memristance is close to the data boundary, because I_{Qm} is nearly equal to I_{Qf}. Metastability can be caused by device mismatch, as illustrated in Fig. 5.10. The regenerative latch is the major source of metastability as the minimum clock periods at 1σ, with and without the current mirror A and DCCS, are not significantly different. The samples with long latency indicate the high impact of the metastable state. Metastability will cause sampling error in synchronous circuits when the clock signal arrives at the receiver, for example flip-flop, during the unstable state [165]. This issue is more significant in the case of asynchronous circuits because the successive stage can step to the incorrect sequence due to metastable input [166].

A metastability resolver (MSR) is attached to Q_m and Q_f to filter out such signals [134]. The circuit implementation is illustrated in Fig. 5.8. When the comparator is in standby mode, logic 0s at Q_m and Q_f turn on MP0 and MP1 so that the inputs of both inverters are 1s (GT and LT are 0s). These PMOSs are still ON during the metastable state. Once the logic becomes fully differential, either MP0/MN1 or MP1/MN0 are ON

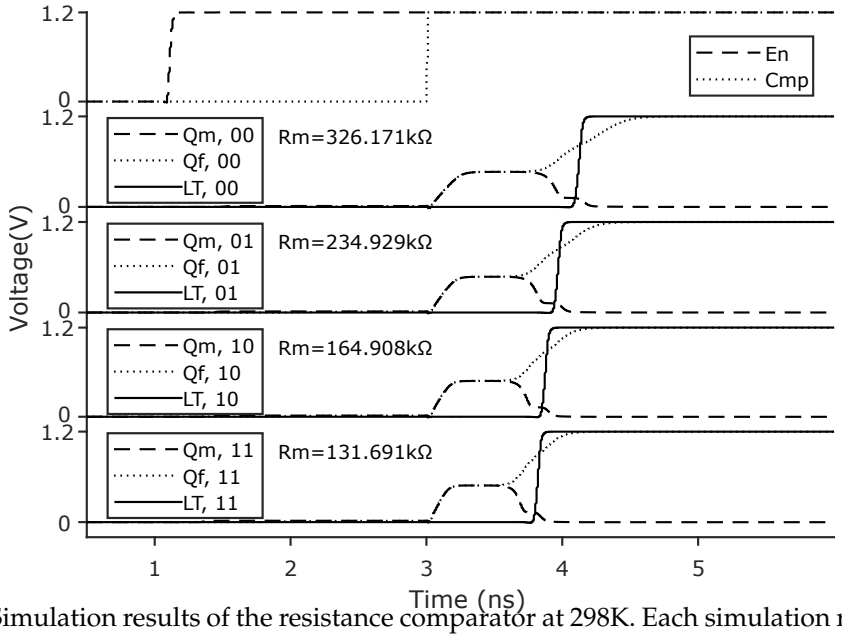


Figure 5.9: Simulation results of the resistance comparator at 298K. Each simulation runs with the effective R_m in Table 5.3 to show the maximum metastability (Q_m and Q_h) which is filtered by the MSR (LT). GT is omitted from all graphs because it stays at 0V throughout the simulation.

accordingly and give the result as listed in Table 5.2.

From the simulation results in Fig. 5.9, memristance is set at the boundary to implement the worst-case scenario, where metastability is maximised since I_{Qf} and I_{Qm} are nearly equal. It shows that the metastable state is longer at lower data because smaller currents are generated (higher memristance). However, such metastability is completely removed by the MSR (GT, LT). Hence, the insertion of the MSR prevents failure in the rest of the system that is caused by this ambiguous state.

The memristor also encounters the metastability issue where the memristance decays after applying programming pulses [126]. This causes data loss and needs a compensation which may sophisticate the circuit design and is subject to future work. To avoid this metastability, a sufficient programming current may be applied. Cheng *et al.* reported that the memristances of their $Ag/SiO_2/Pt$ memristors are stable after using a programming current of 200μA [167].

5.5 Simulation Results

Three simulations determined the offset, latency and energy, and thermal effect of our design. All circuits are implemented using UMC 65nm low-leakage CMOS technology.

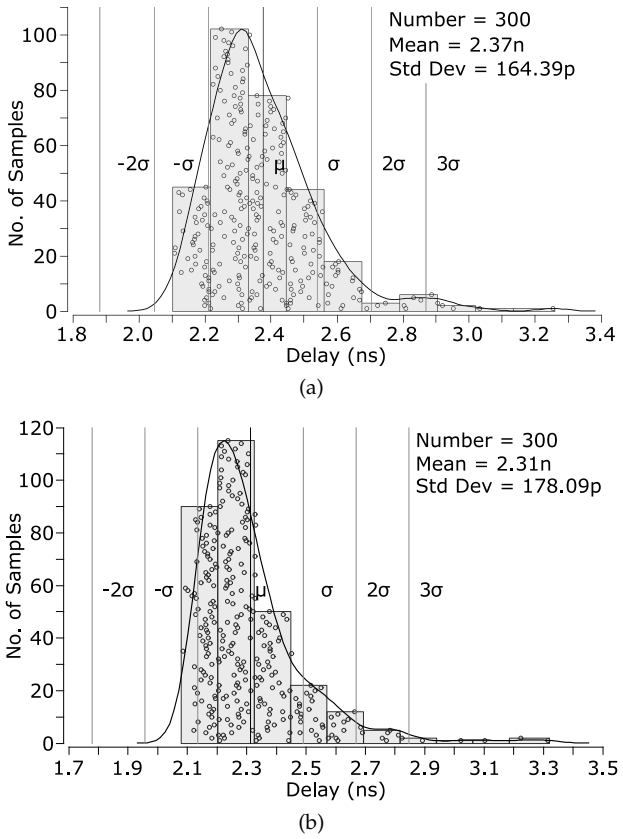


Figure 5.10: Monte Carlo simulations (300 samples) revealing the latency distributions of the resistance comparator: (a) without and (b) with the current mirror A and DCCS. If the variability in the current mirror A and DCCS is not counted, the clock period must not less than 2.54ns to cover the yield at 1σ. Otherwise, the required clock period is slightly changed to 2.48ns. This confirms the major source of metastability is the regenerative latch.

The VTEAM parameters, which were obtained from a practical device [109], and our extracted parameters are listed in Table 4.4. All simulations are conducted using Cadence Spectre.

5.5.1 Offset

Although the DCCS has been tuned to deliver the same amount of currents as expected memristances (boundaries) in Table 5.3, the difference in settling time between I_m and I_f, due to the difference in parasitic capacitance between the current mirror A and the DCCS, still causes offset. To measure the offset, the comparison process is performed with the expected memristance in Table 5.3. Then, the memristance is swept until the effective memristance which actually causes the change in the latch outputs is found. The difference between the two values is considered to be the offset. Fig. 5.11 shows that, at

298K, the offset is positive when the data is at 00 and negative otherwise. The maximum absolute offset is 2.853kΩ (data 10) which is only 1.70% of the expected memristance (167.761kΩ).

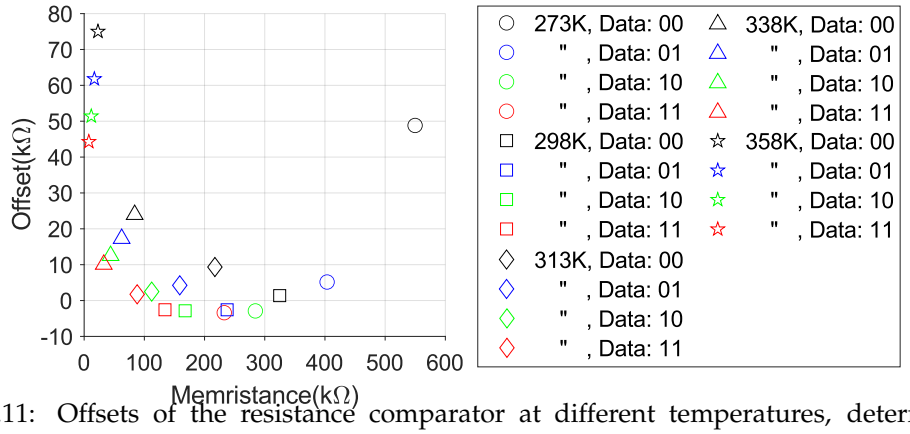


Figure 5.11: Offsets of the resistance comparator at different temperatures, determined by subtracting the expected R_m from the effective one (Table 5.3).

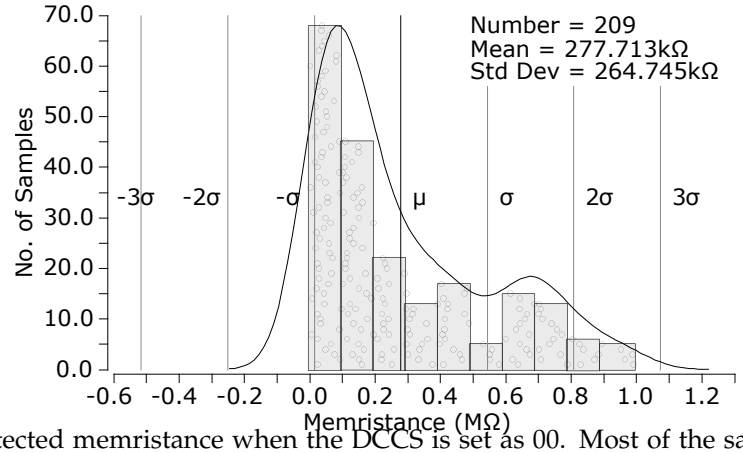


Figure 5.12: Detected memristance when the DCCS is set as 00. Most of the samples fall below the desired range, which is 00. This can be solved by using the offset calibration techniques such as programmable capacitor arrays.

The same procedure is repeated to determine the offset at different temperatures, as illustrated in Fig. 5.11. The absolute offset is less than 10kΩ when the memristor operates between 298K-313K. Otherwise the offset will increase, especially at 273K and 358K. This is because the high memristance at 273K causes insufficient voltage in the connected current mirror. In addition, transistor characteristics at 273K and 358K are highly deviated from the nominal value (298K).

The offset is also caused by process variation. Fig. 5.12 shows the equivalent memristances that generate the same level of current as the DCCS (data 00). It reveals

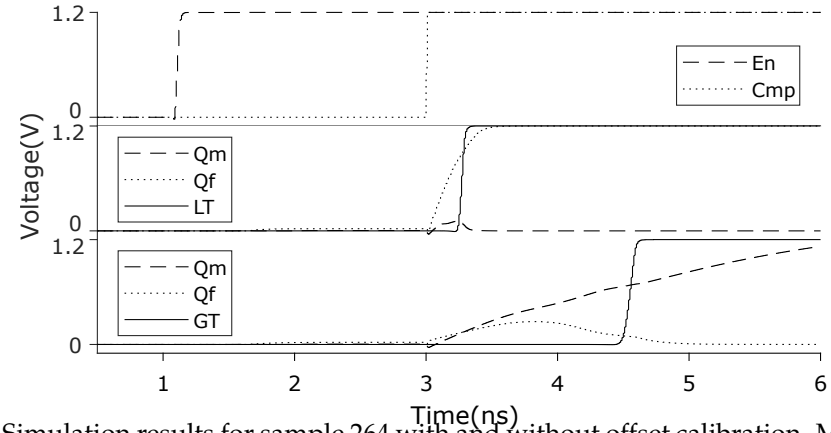


Figure 5.13: Simulation results for sample 264 with and without offset calibration. Memristance is set as 280kΩ, which refers to the data 00. The DCCS is set at 00 which delivers a lower current than the selected memristance. Therefore, the expected result is the logic high at Q_m. However, the middle graph shows that Q_m is low instead. After enabling C_f = 10fF, the signal is compensated and the correct result is shown in the bottom graph.

that variation severely impacts the circuit because the range of memristance is changed from 324.801kΩ-237.506kΩ (Table 5.3) to 0-100kΩ. To solve this problem, digitally programmable capacitor arrays can be attached to the latch outputs [168, 169]. The capacitors C_m and C_f have been connected to Q_m and Q_f through N4 and N5 in Fig. 5.6 to demonstrate this technique. The simulation is based on a value of memristance of 280kΩ, and the Monte Carlo sample 264 (-324.328kΩ offset) without the compensation capacitances is illustrated in the middle graph of Fig. 5.13. The output of the circuit is LT because I_m is less than I_f due to the negative offset. To calibrate this offset, C_f is increased with the step size of 1fF using the parametric sweep function in Cadence ADE. As depicted in the lower graph of Fig. 5.13, the result is correct (GT is raised instead) once C_f reaches 10fF.

5.5.2 Latency and Energy

The latency of the comparison process depends on the memristance. The result takes the longest time (worst) when the effective values of I_{Qm} and I_{Qf} are nearly equal (maximum metastability). Therefore, the worst-case latency is measured at the effective R_m in Table 5.3. On the other hand, latency is shorter once the two currents differ. The best-case latency is defined when the memristance is midway between two adjacent data boundaries. For example, the memristance of 280.550kΩ achieves the best-case latency for data 00.

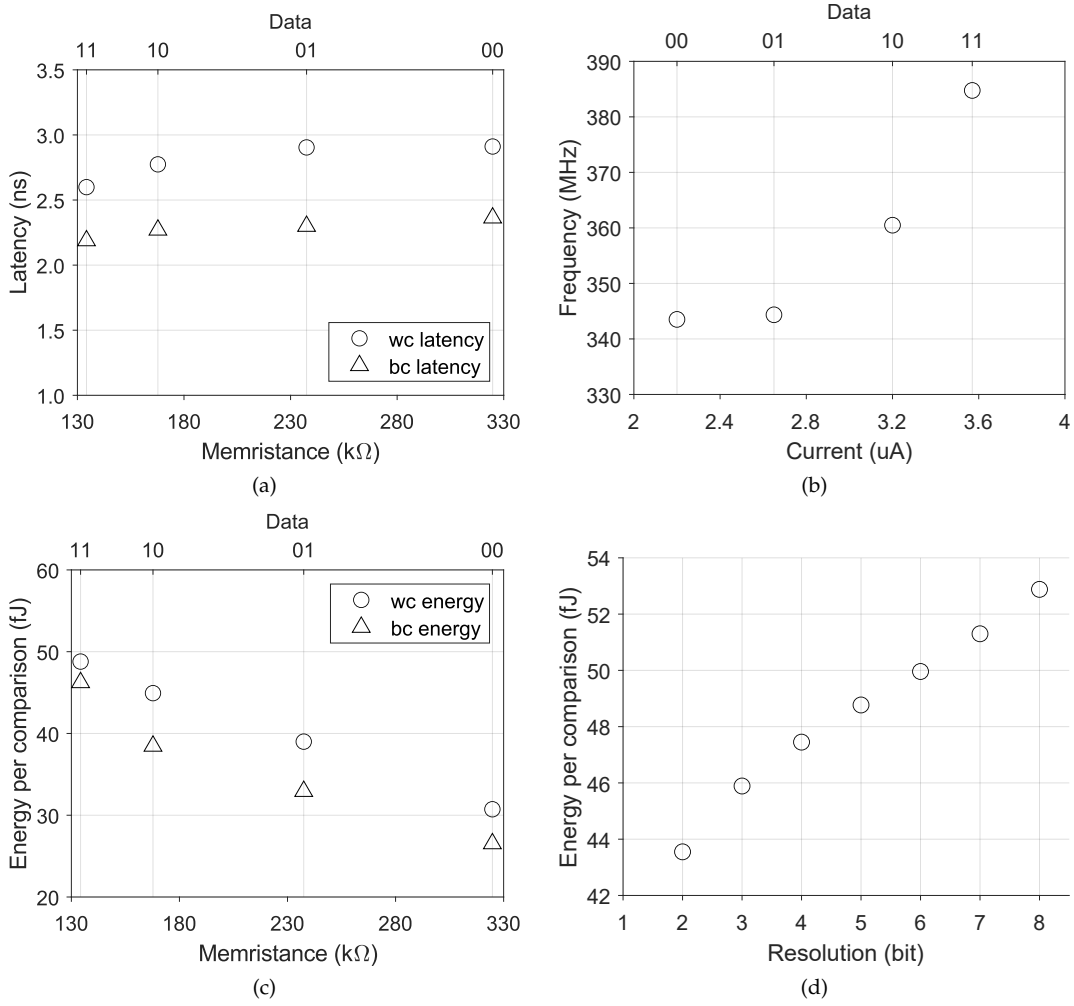


Figure 5.14: Performance of the resistance comparator: (a) the highest latency (2.91ns) is at 00 which has the highest memristance and thus the lowest current; (b) the maximum frequency increases in the same way as the latch input currents; (c) the worst-case energy is higher than the best case for all data because metastability causes a longer comparison time; (e) the maximum energy per comparison for the largest data is higher when the resolution increases, due to lower memristance.

Latency is taken from the transition of En to the output of the MSR (GT/LT). Note that the lower bound of data 11 is selected as 51.691kΩ to maintain the step size of 80kΩ, as explained in Section 5.3. From Fig. 5.14a, the latency of each case slightly decreases with memristance as higher current is generated. The best-case latency (<2.36ns) is approximately 550ps faster than the worst case (<2.91ns). The worst-case latency limits the maximum frequency. It depends on the input currents as depicted in Fig. 5.14b. The input current and frequency at each data boundary increases when the memristance (data) declines. Regarding the worst-case latency, the maximum frequency is calculated

as 343MHz.

The energy of the entire circuit is measured from the rising edge of the En signal to the rising edge of the MSR's output (GT/LT). The results in Table 5.3 show that the circuit's energy in the worst case varies between 30.73fJ to 48.79fJ. From the plot in Fig. 5.14c, the energy rises in opposition to the memristance. This is because the lower memristance draws more current. The energy of the worst-case scenario is higher than in the best case due to the longer latency caused by metastability.

The maximum energy per comparison at each resolution is depicted in Fig. 5.14d. The energy increases due to the higher number of active reference sources and the use of low memristance to support higher resolution. In addition, higher resolution requires a greater number of comparisons, which results in higher power consumption. Thus, reading accuracy can be determined according to the energy available. For example, if the decoder is powered from a stable power supply such as a battery, high precision (8-bit) can be selected; whereas if the decoder is working using energy harvesting, low-energy precision (2-bit) can be selected instead. Note that the comparator's offset must be minimised for high-precision applications. This requires an in-depth offset analysis which remains open for future research.

5.5.3 Temperature effect and compensation

Temperature affects the data ranges in Table 5.3, as illustrated in Fig. 5.15. The ranges are wider at low temperature and become narrow when the temperature rises. The highly compressed data ranges at high temperature, especially over 338K, may need an extremely high precision decoder; otherwise, the IoT devices may not work correctly. For higher memristance values, the effect of temperature on the linearity of the measured memristance is greater. Hence, smaller data values (higher memristance) are more severely affected by temperature variation. This agrees with the modified model in equation (4.8), since the portion of R_{off}, which is highly temperature-dependent and non-linear is large with small data (large memristance).

The temperature also causes decoding error because the memristance and the comparator's references change at different rates. The example in Fig. 5.16 shows that, initially, the memristance is specified as 300kΩ at 298K. Therefore, this memristance stands in the range of 00 (Table 5.3). However, it drops to 152.596kΩ which is decoded

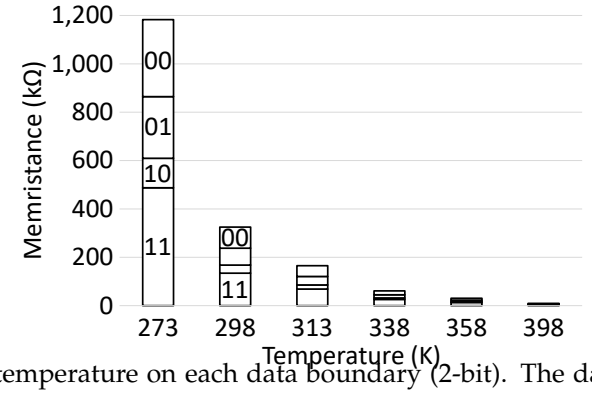


Figure 5.15: Effect of temperature on each data boundary (2-bit). The data ranges are narrower when the temperature rises. They are likely to reach the same point once the temperature is at 358K. This is correlated to the proposed model simulation in Fig. 4.3.

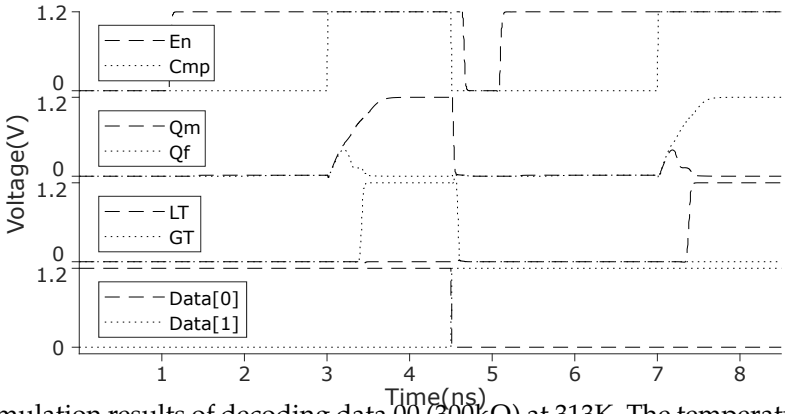


Figure 5.16: Simulation results of decoding data 00 (300kΩ) at 313K. The temperature changes R_m to 152.596kΩ, which is decoded as 10 instead.

as 10 once the temperature rises to 313K. Compensation for the temperature effect is essential in devices working in different temperature conditions.

Compensation can be achieved by calculating the state variables of the upper and lower bounds of the decoded data using equation (2.1). Then, the range between both state variables is mapped to the base range in Table 5.3 to determine the actual data. For example, the decoded data in the previous paragraph is bounded between 131.691kΩ and 164.908kΩ (see Table 5.3). Using R_{off} value at 313K (174.195kΩ), those values can be converted to the state variable of 0.96 and 0.77 respectively. This range falls between the boundaries of data 00. Therefore, the compensated data of 00 will be given at the output of the controller.

Even if the data is compensated correctly in the previous example, there is still an issue where multiple data ranges are condensed such that they stand in the same range at higher temperatures. For instance, suppose that the decoder is designed based on the

range at 298K. When the temperature rises to 313K, the ranges of data 01, 10 and 11 are condensed within the range of 11 at 298K in Fig. 5.15. As a result, the controller cannot determine the difference between them and cannot compensate the data accordingly. A more accurate temperature compensation technique will be investigated in future work.

5.6 Discussion

Because no resistance decoder has yet been reported, the performance of the resistance comparator, which is the largest building block, is instead considered and compared with designs in the literature. The features of the relevant designs are listed in Table 5.4. For the proposed comparator, its performance results shown in the table are selected from the worst results in Table 5.3. Furthermore, the power consumption is calculated by dividing the energy by the latency (time) of comparing data 11, which yields the highest value. The maximum energy-delay product (EDP) of our comparator is also found when comparing data 11.

Table 5.4: Comparison of the specifications of the proposed resistance comparator

Work	Volt. /Cur.	Static /Dyn.	Res. inc.	Tech. (nm)	Volt. (V)	Latency (ns)	Energy (fJ)	EDP ($J \cdot s \times 10^{-24}$)	Power (uW)
[170]	V	S	N	65	1.0	0.20	N/A	N/A	95.00
[171]	V	D	N	180	1.2	14.97	147.00	2,201	N/A
[172]	V	D	N	180	1.2	1.84	N/A	N/A	18.60
[173]	V	D	Y	65	1.2	0.22	760.00	167	755.00
[174]	C	S	N	350	1.0	15.00	N/A	N/A	30.00
[175]	C	S	N	180	1.0	133.00	18.00	2,394	0.14
[176]	C	D	Y	180	0.5	2.20	N/A	N/A	79.00
[136]	C	D	N	180	1.8	0.95	N/A	N/A	697.00
This	C	D	N	65	1.2	2.91	48.79	127	21.11

Although Kim *et al*.'s comparator [170] is faster than ours, it spends more power due to its static design. Therefore, it is not suitable for low-power applications. This is a common issue with any static comparators, such as those proposed by Molinar Solis *et al*. [174] and Suriyavejwongs *et al*. [175].

The performance of Xin *et al*.'s design [172] is prominent among dynamic voltage comparators. It is also faster than our work while consuming less power. However, the bulk input design causes a leakage current to the predecessor stage. In addition, the design by Zhong *et al*. [171] has approximately ten times longer delay, while

Nasrollahpour and S. Hamedi-Hagh's [173] requires a large area due to the use of resistors.

Static current comparators which indicate the direction of the input current have been proposed [174, 175], where one study [174] employs a flipped voltage follower to lower the input impedance while another [175] supports near-threshold operation with zero input offset. Nevertheless, their latencies are very long, especially in Molinar Solis *et al*.'s work [174]. Even though their power consumption figures are extremely low, they still exhibit power losses during idle mode due to static operation. Furthermore, the power demands of their extra circuitry, such as current subtractors, are not included.

Another approach is to translate current into voltage using a differential transimpedance preamplifier (DTIA), which provides better noise reduction according to the differential input design [136, 176]. Jankatkit and V. Kasemsuwan's design [176] is slightly faster than ours, but it has approximately four-fold power requirements. That design does not require extra circuitry, such as current subtractors, and can also operate at near-threshold voltage. However, it contains many large resistors which contributes area overheads and it is prone to temperature and process variations. The design by Sarkar and Banerje [136] also implements DTIA. It is three times faster but needs 35 times higher power due to its static sub-circuits such as the subtractor and amplifiers. Note that the DTIAs in both studies always consume power because of their static operation.

The proposed decoder is suitable for memristive IoT applications as its EDP outperforms those in the literature which require extra circuitry to interface the memristor, control power and support power-accuracy scalability. Furthermore, it can receive current-based input, support temperature compensation and provide a flexible choice of speed and power trade-off, as the sizes of the transistors in the current mirrors (N15 in Fig. 5.7) can be either increased for speed improvement (higher currents) or decreased for power savings (lower currents). It offers scalability in which resolution can be configured to match the power budget. It also supports both synchronous and asynchronous circuits because the operation of the comparator can be controlled by a clock or event-driven signal. This design uses no resistors, which helps to reduce the chip area and also makes it immune to resistivity variations. In cases of systems with multiple memristors (e.g. the crossbar in Fig. 5.1), using a multiplexer can minimise chip area as the devices can be read by a single decoder [90].

5.7 Summary

We propose a memristance decoder, which is based on a current-mode dynamic comparator to support power-efficient and adaptive operation, for memristance reading. Based on UMC 65nm low-leakage CMOS technology, the simulation results of a 2-bit memristance decoder demonstrate its operation with a maximum offset of 1.70%, worst-case latency of 2.91ns, maximum frequency of 343MHz and energy per comparison of 48.79fJ. We show that the metastability issue can be filtered using a metastability resolver. We also show the offset caused by the process variation and its calibration using a configurable capacitor array.

We address challenge (ii) by proposing a memristance decoder that is energy-efficient and supports power-restricted IoT applications, since its resolution can be scaled based on the energy budget. It features our temperature model explained in the previous chapter to improve temperature awareness, so that changes in memristance due to temperature fluctuations can be compensated for. Its current mode design is compatible with a wide range of memristive circuits, from biosensors to machine learning, and also supports both synchronous and asynchronous schemes. Coupling the thermal model with the decoder design for the sensing application and investigating the temperature compensation technique are subjects for future work.

Chapter 6

Conclusion

6.1 Contributions

In this research, memristor-based design solutions have been proposed for the mitigation of parametric variation and improvements in energy savings in IoT devices given two main challenges: (i) enhancing the energy efficiency of the delay element (DE) using memristors; and (ii) improving the temperature awareness and energy robustness of the memristance decoder.

For the first challenge, a memristor was employed in designing a delay element (MemDE) so as to harness its advantages of tunability and non-volatility. Since the internal state of the memristor is in the form of resistance (memristance), placing the memristor between two inverters can contribute an RC delay. The unintended tuning (UT) which is caused by the high voltage drop across the memristor during signal transition has been identified. Adding a flushing transistor is one solution to this problem, but it causes a great difference between rising and falling delays. The simulation results based on AMS 0.35μm HV technology and the VTEAM model with ferroelectric memristor parameters reveal the circuit characteristics, including an effective delay range of 5.48ns to 13.54ns within 6 tuning steps, an average delay of 1.34ns per step, and a minimum tuning pulse width of 3ns.

The memristor-based delay element design (MemDE-UTA) which can avoid UT while maintaining approximately equal rising and falling delays was further investigated. By

re-routing the signal path between rising and falling transitions, this second design can achieve our goal successfully. The simulation results based on UMC 180nm technology and the VTEAM model with the fitting parameters for titanium dioxide memristors reveal the circuit characteristics, including an effective delay range of 0.55ns to 1.44ns within 25 and 47 steps for tuning up and down respectively. They also show average delays for tune up and down operations of 36ps and 19ps per step respectively, and a minimum tuning pulse width of 200ps. For the same technology, our delay range is similar to that of the MuxDE, while our design is more power-efficient as it consumes thirteen times less power.

We also analysed an excitation time model based on the VTEAM model and voltage divider equation to estimate the pulse width for memristance tuning. The general-purpose memristance tuning (GMET) circuit described in the literature, which simplifies memristive circuits, was proposed for this analysis. The simulation results based on AMS 0.35um technology and the VTEAM model with BCM memristor parameters show that our model can estimate the memristance shift regarding the tuning pulse with a maximum error of 7% when memristance dominates the transistor channel resistance ($>$2.5kΩ).

Totally, our MemDE-UTA can be used to adjust the delay to solve the timing violation caused by parametric variations, while also consuming significantly low power. As the delay state is retained in the memristor, no power is needed to operate the external memory devices and to initialise the delay every time the system starts. For these reasons, our MemDE-UTA is suitable for enhancing both tolerance to parametric variation and the energy efficiency of IoT devices. Our excitation time model can be used to estimate the tuning pulse width of our delay element as well as the other memristive circuits to reduce power dissipation and design complexity regarding memristor state verification. The above work fulfils challenge (i).

For the second challenge, the temperature effect on the memristance was modelled to enable temperature awareness in memristive circuits. The effect on OFF resistance is realised using existing data in the literature for the titanium dioxide memristor. In comparison to that data, our model yields a maximum error of 1.25% and an R-squared value of accuracy of 96%. We extended the model's coverage to the silver-chalcogenide memristor, which is the only off-the-shelf device available. Temperature sweep exper-

iments on the physical chips were conducted to extract the model's parameters. The model and the extracted parameters fit the memristor characteristics with a minimum R-squared value of 88%.

We further designed a memristance decoder to demonstrate the integration of the temperature model, the impact of temperature on the decoding process, and temperature compensation. The simulation results for a 2-bit decoder based on UMC 65nm technology and our modified VTEAM model with the fitting parameters for titanium dioxide memristors show that the energy per comparison ranges between 30.73fJ and 48.79fJ. They also show a maximum offset of 1.70% at 298K, a worst-case latency of 2.91ns, and a maximum frequency of 343MHz. Furthermore, the effect of temperature and its compensation when decoding data 00 at 313K are demonstrated.

In summary, we addressed the second challenge by proposing a mathematical model that accurately estimates the effect of temperature on memristance, which is essential for studies of temperature compensation and sensing. This model was validated using measured data from two memristive devices built from different materials. Therefore, it is expected to cover other memristor types. We further integrated this model with our memristance decoder, which was designed for memristive applications, so as to improve its temperature awareness. Data correction regarding the temperature effect is demonstrated. Our decoder is designed to support energy robustness, where its resolution can be configured based on the present energy budget. In addition, it supports both synchronous and asynchronous schemes and involves no resistors, which are a major source of parametric variations.

6.2 Future work

This thesis outlines possibilities for future research in the energy-efficient mitigation of parametric variation for IoT applications.

The method proposed in Section 3.3 facilitates the design of next-generation IoT devices where power consumption and the effects of PVT variations are minimised. To apply the MemDE-UTA to circuits, the designs of the relevant components such as the controller and error detection register (EDR) shown in Figs. 3.1 and 3.2 need to be developed. The EDR will send the error signal to the controller when it detects changes

in signals within a specific timing window. Implementations of EDRs for synchronous circuits can be found elsewhere [15, 16, 80, 137]. Moreover, an implementation for an asynchronous bundled-data (BD) circuit can be found in a further publication [17]. The controller must be able to generate a tune-up pulse once a timing violation is detected. It may include a counter to be able to tune down the delay when the system is free from timing violations for a specified length of time.

The MemDE-UTA proposed in Section 3.3 is compatible with current fabrication technologies smaller than 180nm, which offer only one core voltage. In real implementations, however, designers must select memristor and transistor widths carefully, because the memristor voltage may exceed its thresholds and consequently cause unintended tuning (UT) if the memristance is too high or the transistors are too wide (with low channel resistances). To satisfy the above conditions, a mathematical model that describes the relationship between memristor voltage, memristance and transistor size is a subject for future work.

To build a digital system with the MemDE-UTAs, controllers and EDRs, a synthesis tool and design flow must be developed in the future. The design flow will provide guidelines in synthesising a digital system from RTL languages such as VHDL and Verilog. The synthesis tool must be able insert those components into the longest path which governs the timing of the system. Examples of the memristive-circuit design flows are available the above-mentioned papers [29, 80]. The work from Hand *et al.* [17] describes the design flow for asynchronous BD circuits. The successful development of the tool will allows designers to simulate and observe the mitigation of variation and energy efficiency of the proposed MemDE-UTA using well-known benchmark circuits such as ISCAS'85 [177], ISCAS'89 [178] and IWLS'05 [179].

Section 3.5.3 reveals the overshoot due to gate-drain leakage, which becomes significant when the circuit is implemented with small technology (180nm). This spiking signal can cause over-tuning which cannot be estimated by our excitation time model. There is an opportunity to improve the accuracy of our model by including the overshoot effect, which has been analysed elsewhere [146] and [147].

The proposed temperature model fits the experimental data between 253K and 378K. At 378K, the silver ions that remain in the undoped region start to influence the memristor's characteristic. Therefore, this effect must be included in our model if target

applications operate above 378K. In addition, the effect of these remaining ions can be re-used to describe the effect of temperature on ON-memristance, because it is also caused by the silver ions.

The results of the fixed temperature experiment described in Section 4.3.2 show that memristance is unstable at 378K. Further investigation and device development should be conducted to overcome stability problems at high temperature.

A memristance decoder that supports resolution scalability based on the available energy has been proposed in Chapter 5. In the future, this design will be extended to create a memristive memory device by adding memristance programming circuitry. Our decoder provides a choice to create a memory device based on either synchronous or asynchronous schemes. As noise effects become a concern when feature size is reduced, developing noise reduction techniques, such as the use of differential current-mode comparators, would be beneficial.

Because the memristor is a temperature-sensitive device, coupling it with our memristance decoder offers an opportunity to build a temperature sensor. Section 5.5.3 shows that the controller may not be able to identify data at high temperature because it refers to multiple memristance ranges. For this reason, a more accurate temperature compensation technique remains an open challenge.

Appendix A

Main publications on the thesis

- Bunnam T, Soltan A, Sokolov D, Yakovlev A. *Pulse controlled memristor-based delay element.* In: 27th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS). 2017, Thessaloniki, Greece.
- Bunnam T, Soltan A, Sokolov D, Yakovlev A. *An Excitation Time Model for General-purpose Memristance Tuning Circuit.* In: IEEE International Symposium on Circuits and Systems (ISCAS). 2018, Florence, Italy.
- Bunnam T, Soltan A, Sokolov D, Maevsky O, Yakovlev A. *Toward Designing Thermally-Aware Memristance Decoder.* In: IEEE Transactions on Circuits and Systems I: Regular Papers. Volume 66. Issue 11. 2019. (Presented in ISCAS 2020 conference)
- Bunnam T, Soltan A, Sokolov D, Maevsky O, Degenaar P, Yakovlev A. *Empirical Temperature Model of Self-Directed Channel Memristor.* In: IEEE SENSORS. 2020, Rotterdam, Netherlands.

Appendix B

Fixed-temperature experimental results

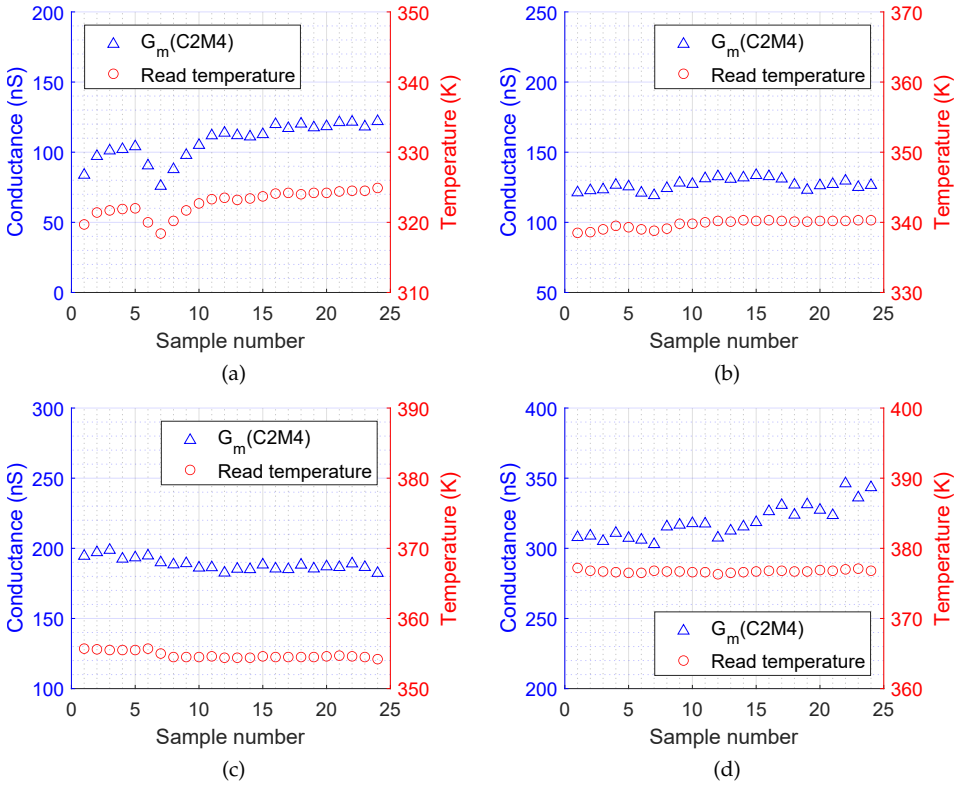


Figure B.1: C2M4's conductance and read temperature vs sample at: (a) 323K; (b) 338K; (c) 353K; (d) 378K.

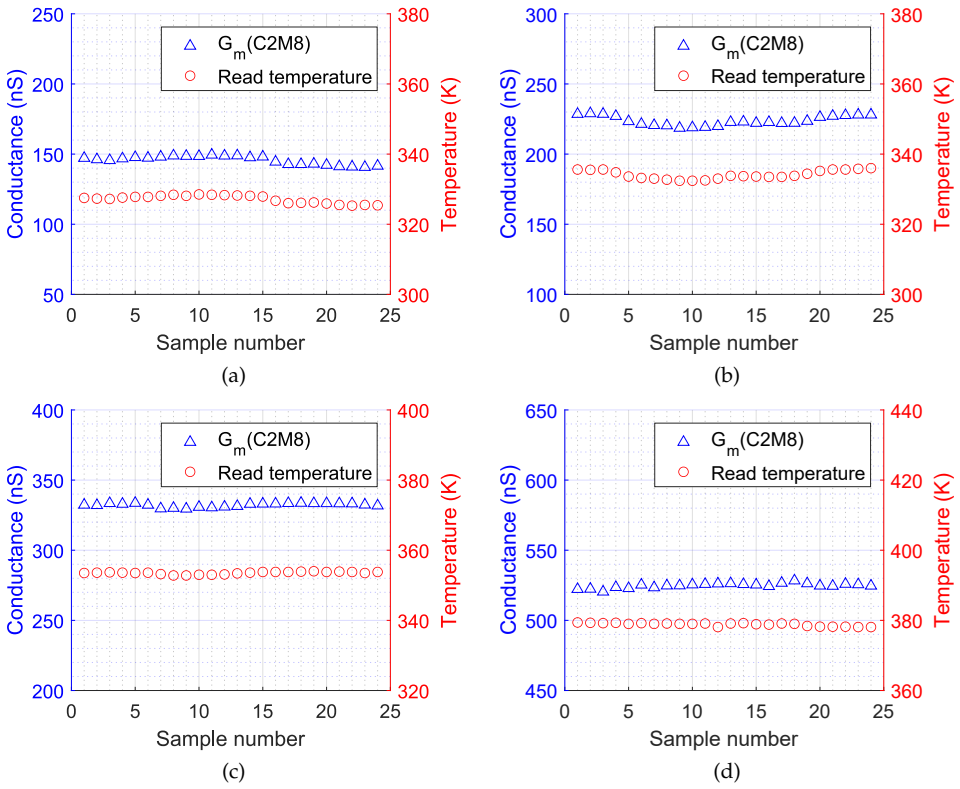


Figure B.2: C2M8's conductance and read temperature vs sample at: (a) 323K; (b) 338K; (c) 353K; (d) 378K.

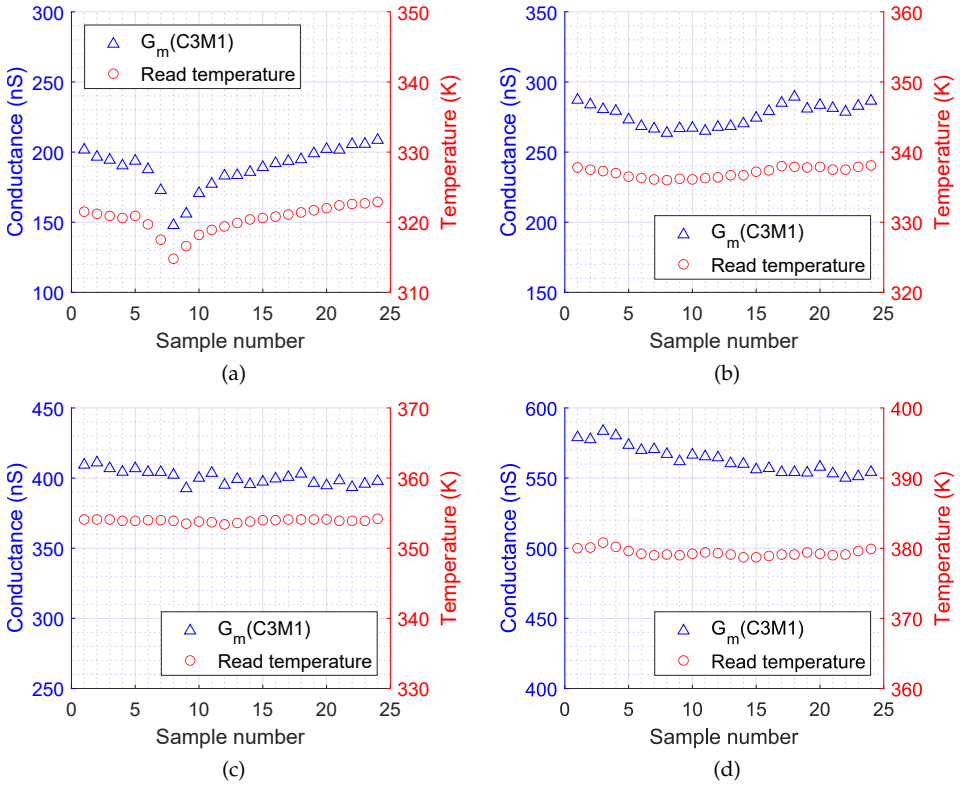


Figure B.3: C3M1's conductance and read temperature vs sample at: (a) 323K; (b) 338K; (c) 353K; (d) 378K.

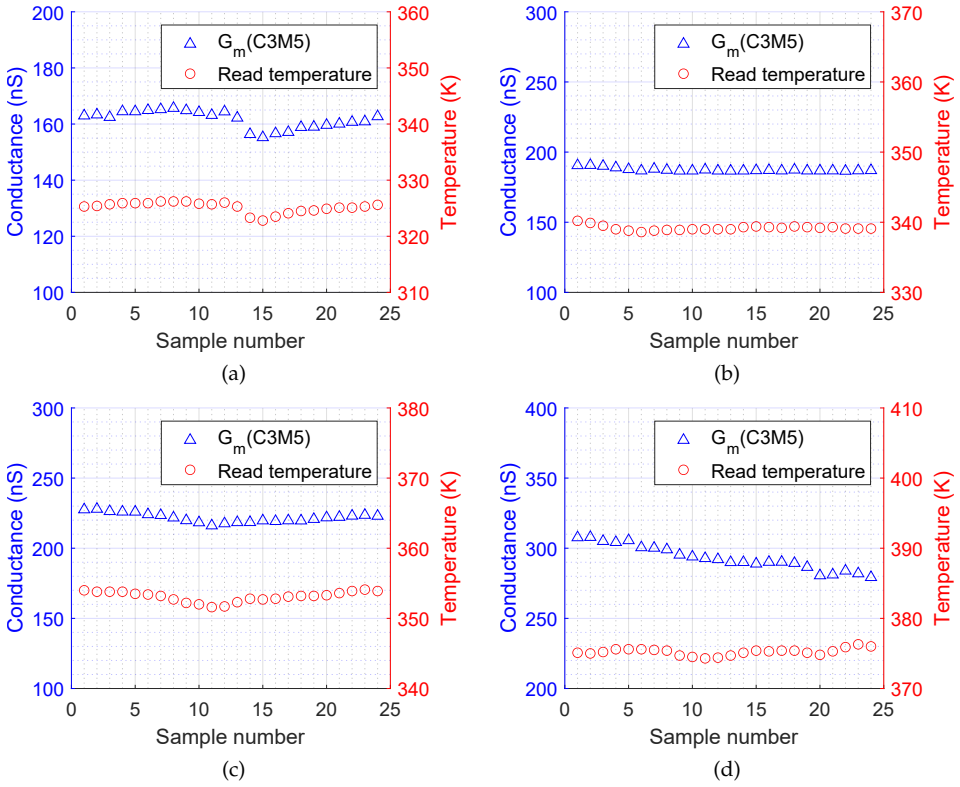


Figure B.4: C3M5's conductance and read temperature vs sample at: (a) 323K; (b) 338K; (c) 353K; (d) 378K.

Appendix C

Verilog-A code of the VTEAM model with integrated temperature effect

```
nature distance
    access = Metr;
    units = "m";
    abstol = 0.01n;
endnature
discipline Distance
    potential distance;
enddiscipline
nature resistance
    access = Res;
    units = "";
    abstol = 0.01n;
endnature
discipline Resistance
    potential resistance;
enddiscipline
module Memristor_tvteam(p, n,w_position, rm);
    inout p, n;
    output rm, w_position; // Rm & w-width pins
    electrical p, n, gnd;
    Distance w_position;
    Resistance rm;
    ground gnd;
    real w_last, stp_multply, first_iteration;
    real x, dxdt, x_last, lambda;
    real p_coeff = 2;
    parameter real rho0 = 0.0122;
    parameter real Ea = 0.36;
    parameter real A  = pow(12e-6, 2);
    parameter real Tk = 298;
    parameter real fix_state = 0; // 0=Rm shifts as usual, 1=fix Rm
    parameter real state_rm  = 0; // specify state by: 0=state, 1=Rm
    parameter real Tk_enable = 0; // temp. response: 0=disable, 1=enable
    parameter real dt        = 1e-12;
    parameter real window = 0; // window fn: 0=disable, 1=enable (Biolek)
    parameter real init_state    = 0;
    parameter real init_state_rm = 1e3;
    parameter real x_off = 3e-9; // equals D
    real kB_eV  = 8.62e-5
    real D      = x_off;
    real Roff   = 300e3;
```

```
42      real Ron    = 1000;
43      real K_on   = -216.2;
44      real K_off  = 91e-3;
45      real Alpha_on   = 4;
46      real Alpha_off  = 4;
47      real v_on   = -1.5;
48      real v_off  = 300e-3;
49      real IV_relation = 0;
50      real x_on   = 0;
51      real Rho, Roff_t, Rho_25, Roff_25;
52      analog function integer stp; //Stp function
53          real arg; input arg;
54          stp = (arg >= 0 ? 1 : 0 );
55      endfunction
56      analog begin
57          if(fix_state==1) first_iteration=0;
58          if(Tk_enable==1) begin // Temperature effect
59              Rho    = rho0*`M_E**(Ea/(kB_eV*Tk));
60              Roff_t = Rho*x_off/A; // Roff regarding temp.
61          end else Roff_t = Roff;
62          Rho_25  = rho0*`M_E**(Ea/(kB_eV*298));
63          Roff_25 = Rho_25*x_off/A; // Roff at 25C
64          if(first_iteration==0) begin
65              if(state_rm==0) begin
66                  w_last = init_state*D;
67                  x_last = init_state*D;
68              end else begin
69                  if(Tk_enable==1) begin
70                  // Fix state, defined by Rm; Specified temp.
71                      w_last = ((init_rm-Ron)/(Roff_t-Ron))*D;
72                      x_last = ((init_rm-Ron)/(Roff_t-Ron))*D;
73                  end else begin
74                  // Fix state, defined by Rm; Fix temp.@25C
75                      w_last = ((init_rm-Ron)/(Roff_25-Ron))*D;
76                      x_last = ((init_rm-Ron)/(Roff_25-Ron))*D;
77          end end end
78          if (V(p,n) >= v_off)
79              dxdt = K_off*pow((V(p,n)/v_off-1),Alpha_off);
80          if (V(p,n) <= v_on)
81              dxdt = K_on*pow((V(p,n)/v_on-1),Alpha_on);
82          if ((v_on<V(p,n)) && (V(p,n)<v_off)) dxdt=0;
83          if (window==0) // No window
84              x = x_last+dt*dxdt;
85          if (window==1) begin // Biolek window
86              if (stp(-V(p,n))==1) stp_multply = 1;
87              if (stp(-V(p,n))==0) stp_multply = 0;
88              x=x_last+dt*dxdt*
89                  (1-pow(pow((x_last/D-stp_multply),2),
90                  p_coeff));
91          end
92          if (x>=D) begin
93              dxdt = 0;
94              x    = D;
95          end
96          if (x<=0) begin
97              dxdt = 0;
```

```
98              x    = 0;
99          end
100         lambda = ln(Roff/Ron);
101         x_last = x; //update the output ports(pins)
102         Metr(w_position) <+ x/D;
103         if (IV_relation==1) begin
104             V(p,n) <+ Ron*I(p,n)*
105                 exp(lambda*(x-x_on)/(x_off-x_on));
106         end else if (IV_relation==0) begin
107             V(p,n) <+ (Roff_t*x/D+Ron*(1-x/D))*I(p,n);
108             Res(rm) <+ Roff_t*x/D+Ron*(1-x/D);
109         end
110         first_iteration = 1;
111 end endmodule
```

Bibliography

[1] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E. Graves, Z. Li, J. P. Strachan, P. Lin, Z. Wang, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, and Q. Xia, "Analogue signal and image processing with large memristor crossbars," *Nature Electronics*, vol. 1, no. 1, pp. 52–59, 2018.

[2] O. Krestinskaya, A. Pappachen James, L. O. Chua, A. P. James, and L. O. Chua, "Neuro-memristive Circuits for Edge Computing: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 1, pp. 4–23, Jan 2020.

[3] C. Yakopcic, R. Hasan, and T. M. Taha, "Memristor based neuromorphic circuit for ex-situ training of multi-layer neural network algorithms," in *Proceedings of the International Joint Conference on Neural Networks*, 2015.

[4] L. Ni, Z. Liu, H. Yu, and R. V. Joshi, "An Energy-Efficient Digital ReRAM-Crossbar-Based CNN With Bitwise Parallelism," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, 2017.

[5] D. Blaauw, D. Sylvester, P. Dutta, Y. Lee, I. Lee, S. Bang, Y. Kim, G. Kim, P. Pannuto, Y.-S. Kuo, D. Yoon, W. Jung, Z. Foo, Y.-P. Chen, S. Oh, S. Jeong, and M. Choi, "IoT design space challenges: Circuits and systems," in *2014 Symposium on VLSI Technology (VLSI-Technology): Digest of Technical Papers*. IEEE, Jun 2014, pp. 1–2.

[6] M. Pelgrom, A. Duinmaijer, and A. Welbers, "Matching properties of MOS transistors," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1439, Oct 1989.

[7] Y. Ye, S. Gummalla, C.-C. Wang, C. Chakrabarti, and Y. Cao, "Random variability modeling and its impact on scaled CMOS circuits," *Journal of Computational Electronics*, vol. 9, no. 3-4, pp. 108–113, Dec 2010.

[8] X. Zhang, D. Shang, F. Xia, H. S. Low, and A. Yakovlev, "A hybrid power delivery method for asynchronous loads in energy harvesting systems," in *10th IEEE International NEWCAS Conference*. IEEE, Jun 2012, pp. 413–416.

[9] W. Neil and H. David, *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison-Wesley Publishing Company, 2010.

[10] S. Ghosh and K. Roy, "Parameter Variation Tolerance and Error Resiliency: New Design Paradigm for the Nanoscale Era," *Proceedings of the IEEE*, vol. 98, no. 10, pp. 1718–1751, Oct 2010.

[11] D. Young and A. Christou, "Failure Mechanism Models for Electromigration," *IEEE Transactions on Reliability*, 1994.

[12] M. A. Scarpato, "Digital circuit performance estimation under PVT and aging effects," Ph.D. dissertation, Micro and nanotechnologies/Microelectronics. Université Grenoble Alpes, Dec 2017.

[13] E. Takahashi, Y. Kasai, M. Murakawa, and T. Higuchi, "Post-fabrication clock-timing adjustment using genetic algorithms," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 4, pp. 643–650, 2004.

[14] H. Mashiko and Y. Kohira, "Yield and power improvement method by post-silicon delay tuning and technology mapping," in *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, 2016, pp. 366–369.

[15] S. Das, C. Tokunaga, S. Pant, W.-H. Ma, S. Kalaiselvan, K. Lai, D. M. Bull, and D. T. Blaauw, "RazorII: In Situ Error Detection and Correction for PVT and SER Tolerance," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 32–48, Jan 2009.

[16] M. Fojtik, D. Fick, Y. Kim, N. Pinckney, D. M. Harris, D. Blaauw, and D. Sylvester, "Bubble Razor: Eliminating Timing Margins in an ARM Cortex-M3 Processor in 45 nm CMOS Using Architecturally Independent Error Detection and Correction," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 66–81, Jan 2013.

[17] D. Hand, M. T. Moreira, H. H. Huang, D. Chen, F. Butzke, Z. Li, M. Gibiluka, M. Breuer, N. L. V. Calazans, and P. A. Beerel, "Blade – a timing violation resilient asynchronous template," in *IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, 2015, pp. 21–28.

[18] P. A. Beerel, R. O. Ozdag, and M. Ferretti, *A Designer's Guide to Asynchronous VLSI*. Cambridge: Cambridge University Press, 2010.

[19] J. Sparsø, *Introduction to Asynchronous Circuit Design*. DTU Compute, Technical University of Denmark, 2020.

[20] S. Nowick, "Design of a low-latency asynchronous adder using speculative completion," *IEE Proceedings - Computers and Digital Techniques*, vol. 143, no. 5, p. 301, 1996.

[21] W. Dally, "Challenges for Future Computing Systems," in *HiPEAC Keynote*, 2015.

[22] L. Chua, "Memristor-the missing circuit element," *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971.

[23] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.

[24] J. J. Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," *Nat Nano*, vol. 8, no. 1, pp. 13–24, 2013.

[25] V. Rana and R. Waser, "Redox-Based Memristive Devices," in *Memristors and Memristive Systems*, R. Tetzlaff, Ed. New York, NY: Springer New York, 2014, pp. 223–251.

[26] I. Vourkas and G. C. Sirakoulis, "Memristive Crossbar-Based Nonvolatile Memory," in *Memristor-Based Nanoelectronic Computing Circuits and Architectures*. Cham: Springer International Publishing, 2016, pp. 101–147.

[27] S. Stathopoulos, A. Khiat, M. Trapatseli, S. Cortese, A. Serb, I. Valov, and T. Prodromakis, "Multibit memory operation of metal-oxide bi-layer memristors," *Scientific Reports*, vol. 7, no. 1, p. 17532, 2017.

[28] H. Abunahla and B. Mohammad, "Memristor Device Overview," in *Memristor Technology: Synthesis and Modeling for Sensing and Security Applications*. Cham: Springer International Publishing, 2018, pp. 1–29.

[29] S. Maheshwari, S. Stathopoulos, J. Wang, A. Serb, Y. Pan, L. B. Leene, C. Papavassiliou, T. G. Constandinou, and T. Prodromakis, "Hybrid CMOS/Memristor Circuit Design Methodology," *arXiv e-prints*, p. arXiv:2012.02267, Dec 2020.

[30] R. Shafik, A. Yakovlev, and S. Das, "Real-Power Computing," *IEEE Transactions on Computers*, vol. 67, no. 10, pp. 1445–1461, Oct 2018.

[31] N. Wald and S. Kvatinsky, "Understanding the influence of device, circuit and environmental variations on real processing in memristive memory using Memristor Aided Logic," *Microelectronics Journal*, vol. 86, pp. 22–33, Apr 2019.

[32] M. Hu, H. Li, Y. Chen, X. Wang, and R. E. Pino, "Geometry variations analysis of TiO2 thin-film and spintronic memristors," in *16th Asia and South Pacific Design Automation Conference (ASP-DAC 2011)*. IEEE, Jan 2011, pp. 25–30.

[33] Q. Li, H. Xu, A. Khiat, Z. Sun, and T. Prodromakis, "Impact of active areas on electrical characteristics of TiO_2 based solid-state memristors," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015, pp. 185–188.

[34] S. Kvatinsky, M. Ramadan, E. G. Friedman, and A. Kolodny, "VTEAM: A general model for voltage-controlled memristors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 8, pp. 786–790, 2015.

[35] R. G. Breckenridge and W. R. Hosler, "Electrical Properties of Titanium Dioxide Semiconductors," *Physical Review*, vol. 91, no. 4, pp. 793–802, Aug 1953.

[36] A. Yildiz, S. B. Lisesivdin, M. Kasap, and D. Mardare, "Electrical properties of TiO2 thin films," *Journal of Non-Crystalline Solids*, vol. 354, no. 45-46, pp. 4944–4947, Nov 2008.

[37] R. Waser, R. Dittmann, G. Staikov, and K. Szot, "Redox-Based Resistive Switching Memories - Nanoionic Mechanisms, Prospects, and Challenges," *Advanced Materials*, vol. 21, no. 25-26, pp. 2632–2663, Jul 2009.

[38] S. Kugler and K. Shimakawa, *Amorphous Semiconductors*. Cambridge: Cambridge University Press, 2015.

[39] K. Morigaki and C. Ogihara, "Amorphous Semiconductors: Structure, Optical, and Electrical Properties," in *Springer Handbook of Electronic and Photonic Materials*. Cham: Springer International Publishing, 2017, pp. 1–1.

[40] R. B. Nuernberg, "Numerical comparison of usual Arrhenius-type equations for modeling ionic transport in solids," *Ionics*, vol. 26, no. 5, pp. 2405–2412, May 2020.

[41] K. A. Campbell, "Self-directed channel memristor for high temperature operation," *Microelectronics Journal*, vol. 59, pp. 10–14, 2017.

[42] H. Abunahla, B. Mohammad, and D. Al Homouz, "Effect of device, size, activation energy, temperature, and frequency on memristor switching time," in *2014 26th International Conference on Microelectronics (ICM)*. IEEE, Dec 2014, Conference Proceedings, pp. 60–63.

[43] M. Hu, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, J. J. Yang, and R. S. Williams, "Dot-product engine for neuromorphic computing: programming 1T1M crossbar to accelerate matrix-vector multiplication," in *Annual design automation Conference*. ACM, 2016, p. 19.

[44] S. Yu, A. Soltan, R. Shafik, T. Bunnam, F. Xia, D. Balsamo, and A. Yakovlev, "Current-mode carry-free multiplier design using a memristor-transistor crossbar architecture," in *Design, Automation Test in Europe Conference and Exhibition (DATE)*, 2020, pp. 638–641.

[45] H. Mostafa, A. Khiat, A. Serb, C. G. Mayr, G. Indiveri, and T. Prodromakis, "Implementation of a spike-based perceptron learning rule using TiO(2-x) memristors," *Frontiers in Neuroscience*, vol. 9, p. 357, 2015.

[46] I. Gupta, A. Serb, A. Khiat, R. Zeitler, S. Vassanelli, and T. Prodromakis, "Real-time encoding and compression of neuronal spikes by metal-oxide memristors," *Nature Communications*, vol. 7, no. 1, p. 12805, Nov 2016.

[47] A. Serb, J. Bill, A. Khiat, R. Berdan, R. Legenstein, and T. Prodromakis, "Unsupervised learning in probabilistic neural networks with multi-state metal-oxide memristive synapses," *Nature Communications*, vol. 7, no. 1, p. 12611, Nov 2016.

[48] I. Tzouvadaki, F. Puppo, M. A. Doucey, G. D. Micheli, and S. Carrara, "Computational study on the electrical behavior of silicon nanowire memristive biosensors," *IEEE Sensors Journal*, vol. 15, no. 11, pp. 6208–6217, 2015.

[49] F. Puppo, F. L. Traversa, M. D. Ventra, G. D. Micheli, and S. Carrara, "Surface trap mediated electronic transport in biofunctionalized silicon nanowires," *Nanotechnology*, vol. 27, no. 34, p. 345503, 2016.

[50] B. Ibarlucea, T. Fawzul Akbar, K. Kim, T. Rim, C.-K. Baek, A. Ascoli, R. Tetzlaff, L. Baraban, and G. Cuniberti, "Ultrasensitive detection of Ebola matrix protein in a memristor mode," *Nano Research*, vol. 11, no. 2, pp. 1057–1068, 2018.

[51] H. Manem and G. S. Rose, "A read-monitored write circuit for 1T1M multi-level memristor memories," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2011, pp. 2938–2941.

[52] A. Emara, M. Ghoneima, and M. El-Dessouky, "Differential 1T2M memristor memory cell for single/multi-bit RRAM modules," in *Computer Science and Electronic Engineering Conference (CEEC)*, 2014, pp. 69–72.

[53] X. Wang, S. Li, H. Liu, and Z. Zeng, "A compact scheme of reading and writing for memristor-based multi-valued memory," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[54] Y. Yilmaz and P. Mazumder, "A drift-tolerant read/write scheme for multi-level memristor memory," *IEEE Transactions on Nanotechnology*, vol. PP, no. 99, pp. 1–1, 2017.

[55] U. Kovac, D. Reid, C. Millar, G. Roy, S. Roy, and A. Asenov, "Statistical simulation of random dopant induced threshold voltage fluctuations for 35nm channel length MOSFET," *Microelectronics Reliability*, vol. 48, no. 8-9, pp. 1572–1575, Aug 2008.

[56] D. Reid, C. Millar, G. Roy, S. Roy, and A. Asenov, "Analysis of Threshold Voltage Distribution Due to Random Dopants: A 100 000-Sample 3-D Simulation Study," *IEEE Transactions on Electron Devices*, vol. 56, no. 10, pp. 2255–2263, Oct 2009.

[57] S. Kundu, A. Sreedhar, and A. Sanyal, "Forbidden pitches in sub-wavelength lithography and their implications on design," *Journal of Computer-Aided Materials Design*, vol. 14, no. 1, pp. 79–89, Apr 2007.

[58] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *Proceedings of the 40th conference on Design automation - DAC '03*. New York, New York, USA: ACM Press, 2003, p. 338.

[59] F. M. Khoshnaw, "Evaluation the moisture effects on the performance of electronic devices," in *2010 IEEE 14th Workshop on Signal Propagation on Interconnects*. IEEE, 2010, pp. 59–61.

[60] F. Stellari, C. Cabral, P. Song, and R. Laibowitz, "Humidity Penetration Impact on Integrated Circuit Performance and Reliability," in *2019 IEEE International Electron Devices Meeting (IEDM)*. IEEE, Dec 2019, pp. 5.4.1–5.4.4.

[61] C. Eder, V. Valente, N. Donaldson, and A. Demosthenous, "A CMOS Smart Temperature and Humidity Sensor with Combined Readout," *Sensors*, vol. 14, no. 9, pp. 17 192–17 211, Sep 2014.

[62] G. Borghello, "Ionizing radiation effects in nanoscale CMOS technologies exposed to ultra-high doses," *University of Udine*, Oct 2018.

[63] J. Prinzie, M. Steyaert, and P. Leroux, "Radiation Effects in CMOS Technology," in *Radiation Hardened CMOS Integrated Circuits for Time-Based Signal Processing*, 1st ed. Cham: Springer, 2018, ch. 1, pp. 1–20.

[64] D. K. Schroder and J. A. Babcock, "Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing," 2003.

[65] V. Huard, M. Denais, and C. Parthasarathy, "NBTI degradation: From physical mechanisms to modelling," *Microelectronics Reliability*, 2006.

[66] X. Guo, W. Burleson, and M. Stan, "Modeling and experimental demonstration of accelerated self-healing techniques," in *Proceedings - Design Automation Conference*, 2014.

[67] D. P. Ioannou, S. Mittl, and G. La Rosa, "Positive bias temperature instability effects in nMOSFETs with HfO 2/TiN gate stacks," in *IEEE Transactions on Device and Materials Reliability*, 2009.

[68] X. Wang, Q. Tang, P. Jain, D. Jiao, and C. H. Kim, "The dependence of BTI and HCI-induced frequency degradation on interconnect length and its circuit level implications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2015.

[69] S. Tyaginov, M. Jech, J. Franco, P. Sharma, B. Kaczer, and T. Grasser, "Understanding and Modeling the Temperature Behavior of Hot-Carrier Degradation in SiON nMOSFETs," *IEEE Electron Device Letters*, 2016.

[70] C. Schlünder, S. Aresu, G. Georgakos, W. Kanert, H. Reisinger, K. Hofmann, and W. Gustin, "HCI vs. BTI? - Neither one's out," in *IEEE International Reliability Physics Symposium Proceedings*, 2012.

[71] M. Alam, B. Weir, and A. Silverman, "A future of function or failure? [CMOS gate oxide scaling]," *IEEE Circuits and Devices Magazine*, vol. 18, no. 2, pp. 42–48, Mar 2002.

[72] R. Ye, F. Yuan, and Q. Xu, "Online clock skew tuning for timing speculation," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, Nov 2011, Conference Proceedings, pp. 442–447.

[73] O. C. Akgun, Y. Leblebici, and E. A. Vittoz, "Current sensing completion detection for subthreshold asynchronous circuits," in *2007 18th European Conference on Circuit Theory and Design*. IEEE, Aug 2007, pp. 376–379.

[74] K. van Berkel, R. Burgess, J. Kessels, A. Peeters, M. Roncken, F. Schalij, and R. van de Wiel, "A single-rail re-implementation of a DCC error detector using a generic standard-cell library," in *Proceedings Second Working Conference on Asynchronous Design Methodologies*. IEEE Comput. Soc. Press, 1995, pp. 72–79.

[75] S. Tugsinavisut, S. Jirayucharoensak, and P. Beerel, "An asynchronous pipeline comparisons with application to DCT matrix-vector multiplication," in *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS '03.*, vol. 5. IEEE, 2003, pp. V–361–V–364.

[76] M. Moreira, B. Oliveira, J. Pontes, and N. Calazans, "A 65nm standard cell set and flow dedicated to automated asynchronous circuits design," in *2011 IEEE International SOC Conference*. IEEE, Sep 2011, pp. 99–104.

[77] P. A. Beerel and M. E. Roncken, "Low Power and Energy Efficient Asynchronous Design," *Journal of Low Power Electronics*, vol. 3, no. 3, pp. 234–253, Dec 2007.

[78] S. A. Aketi, S. Gupta, H. Cheng, J. Mekie, and P. A. Beerel, "SERAD: Soft Error Resilient Asynchronous Design Using a Bundled Data Protocol," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 5, pp. 1667–1677, May 2020.

[79] R. Ozdag and P. Beerel, "An Asynchronous Low-Power High-Performance Sequential Decoder Implemented With QDI Templates," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 9, pp. 975–985, Sep 2006.

[80] J. Gu and J. Li, "Exploration of self-healing circuits for timing resilient design using emerging memristor devices," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015, pp. 1458–1461.

[81] C. Jen-Shiun and C. Kuang-Yuan, "The design of an all-digital phase-locked loop with small DCO hardware and fast phase lock," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 7, pp. 945–950, 1999.

[82] M. Maymandi-Nejad and M. Sachdev, "A monotonic digitally controlled delay element," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 11, pp. 2212–2219, 2005.

[83] J. Tschanz, K. Bowman, S. Walstra, M. Agostinelli, T. Karnik, and V. De, "Tunable replica circuits and adaptive voltage-frequency techniques for dynamic voltage, temperature, and aging variation tolerance," in *Symposium on VLSI Circuits*, 2009, pp. 112–113.

[84] G. Heck, L. S. Heck, A. Singhvi, M. T. Moreira, P. A. Beerel, and N. L. Calazans, "Analysis and Optimization of Programmable Delay Elements for 2-Phase Bundled-Data Circuits," in *28th International Conference on VLSI Design*. IEEE, Jan 2015, Conference Proceedings, pp. 321–326.

[85] A. Seraj, M. Maymandi-Nejad, P. Bahmanyar, and M. Sachdev, "A linear comparator-based fully digital delay element," in *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2015, pp. 652–655.

[86] K. Mohammad, T. Tekeste, B. Mohammad, H. Saleh, and M. Qurran, "Embedded memory options for ultra-low power IoT devices," *Microelectronics Journal*, p. 104634, 2019.

[87] X. Zhang, Z. Ma, J. Yu, and L. Xie, "Memristor-based programmable delay element," in *IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, 2014, pp. 1–3.

[88] S. M. A. B. Mokhtar and W. F. H. W. Abdullah, "Memristor based delay element using current starved inverter," in *IEEE Regional Symposium on Micro and Nanoelectronics (RSM)*, 2013, pp. 81–84.

[89] J. Blanc and D. L. Staebler, "Electrocoloration in SrTiO3: Vacancy Drift and Oxidation-Reduction of Transition Metals," *Physical Review B*, vol. 4, no. 10, pp. 3548–3557, Nov 1971.

[90] C. E. Merkel and D. Kudithipudi, "Towards thermal profiling in cmos/memristor hybrid rram architectures," in *25th International Conference on VLSI Design*, 2012, pp. 167–172.

[91] E. Lehtonen and M. Laiho, "CNN using memristors for neighborhood connections," in *International Workshop on Cellular Nanoscale Networks and their Applications (CNNA)*, 2010, pp. 1–4.

[92] D. P. Matthew, B. S. Dmitri, L. B. Julien, J. J. Yang, S. S. Gregory, R. S. Duncan, and R. S. Williams, "Switching dynamics in titanium dioxide memristive devices," *Journal of Applied Physics*, vol. 106, no. 7, p. 074508, 2009.

[93] K. J. Yoon, G. H. Kim, S. Yoo, W. Bae, J. H. Yoon, T. H. Park, D. E. Kwon, Y. J. Kwon, H. J. Kim, Y. M. Kim, and C. S. Hwang, "Double-Layer-Stacked One Diode-One Resistive Switching Memory Crossbar Array with an Extremely High Rectification Ratio of 10 9," *Advanced Electronic Materials*, vol. 3, no. 7, p. 1700152, Jul 2017.

[94] L. Qingjiang, A. Khiat, I. Salaoru, C. Papavassiliou, X. Hui, and T. Prodromakis, "Memory impedance in TiO2 based metal-insulator-metal Devices," *Scientific Reports*, 2014.

[95] I. Salaoru, A. Khiat, Q. Li, R. Berdan, C. Papavassiliou, and T. Prodromakis, "Origin of the off state variability in ReRAM cells," *Journal of Physics D: Applied Physics*, 2014.

[96] H. Abunahla, B. Mohammad, D. Homouz, and C. J. Okelly, "Modeling valance change memristor device: Oxide thickness, material type, and temperature effects," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 12, pp. 2139–2148, 2016.

[97] C. E. Merkel and D. Kudithipudi, "Temperature Sensing RRAM Architecture for 3-D ICs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 4, pp. 878–887, Apr 2014.

[98] Z. Jiang, Y. Wu, S. Yu, L. Yang, K. Song, Z. Karim, and H. P. Wong, "A compact model for metal-oxide resistive random access memory with experiment verification," *IEEE Transactions on Electron Devices*, vol. 63, no. 5, pp. 1884–1892, 2016.

[99] H. Abunahla, B. Mohammad, L. Mahmoud, M. Darweesh, M. Alhawari, M. A. Jaoude, and G. W. Hitt, "MemSens: Memristor-Based Radiation Sensor," *IEEE Sensors Journal*, vol. 18, no. 8, pp. 3198–3205, Apr 2018.

[100] S. G. Dahl, R. Ivans, and K. D. Cantley, "Modeling Memristor Radiation Interaction Events and the Effect on Neuromorphic Learning Circuits," in *Proceedings of the International Conference on Neuromorphic Systems*. ACM, Jul 2018, pp. 1–8.

[101] B. Garda and Z. Galias, "Modeling Sinusoidally Driven Self-Directed Channel Memristors," in *International Conference on Signals and Electronic Systems (ICSES)*, 2018, Conference Proceedings, pp. 19–22.

[102] T. W. Molter and M. A. Nugent, "The generalized metastable switch memristor model," in *15th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA)*, 2016, Conference Proceedings, pp. 1–2.

[103] C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, "Generalized Memristive Device SPICE Model and its Application in Circuit Design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 8, pp. 1201–1214, 2013.

[104] A. S. Oblea, A. Timilsina, D. Moore, and K. A. Campbell, "Silver chalcogenide based memristor devices," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, Jul 2010, Conference Proceedings, pp. 1–3.

[105] M. R. Latif, P. H. Davis, W. B. Knowton, and M. Mitkova, "CBRAM devices based on a nanotube chalcogenide glass structure," *Journal of Materials Science: Materials in Electronics*, vol. 30, no. 3, pp. 2389–2402, Feb 2019.

[106] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "TEAM: ThrEshold Adaptive Memristor model," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 1, pp. 211–221, 2013.

[107] E. Yalon, A. Gavrilov, S. Cohen, D. Mistele, B. Meyler, J. Salzman, and D. Ritter, "Resistive Switching in HfO_2 Probed by a Metal–Insulator–Semiconductor Bipolar Transistor," *IEEE Electron Device Letters*, vol. 33, no. 1, pp. 11–13, 2012.

[108] C. Ho, S. Chang, C. Huang, Y. Chuang, S. Lim, M. Hsieh, S. Chang, and H. Liao, "Integrated HfO2-RRAM to achieve highly reliable, greener, faster, cost-effective, and scaled devices," in *IEEE International Electron Devices Meeting (IEDM)*, 2017, Conference Proceedings, pp. 2.6.1–2.6.4.

[109] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "MAGIC-Memristor-Aided Logic," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, pp. 895–899, 2014.

[110] N. Talati, S. Gupta, P. Mane, and S. Kvatinsky, "Logic Design Within Memristive Memories Using Memristor-Aided loGIC (MAGIC)," *IEEE Transactions on Nanotechnology*, vol. 15, no. 4, pp. 635–650, Jul 2016.

[111] A. Chanthbouala, V. Garcia, R. O. Cherifi, K. Bouzehouane, S. Fusil, X. Moya, S. Xavier, H. Yamada, C. Deranlot, N. D. Mathur, M. Bibes, A. Barthélémy, and J. Grollier, "A ferroelectric memristor," *Nature Materials*, vol. 11, no. 10, pp. 860–864, 2012.

[112] S. Pi, M. Ghadiri-Sadrabadi, J. C. Bardin, and Q. Xia, "Nanoscale memristive radiofrequency switches," *Nature Communications*, vol. 6, no. 1, p. 7519, Nov 2015.

[113] Z. Kolka, J. Vavra, V. Biolkova, A. Ascoli, R. Tetzlaff, and D. Biolek, "Programmable Emulator of Genuinely Floating Memristive Switching Devices," in *26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2019, Conference Proceedings, pp. 217–220.

[114] N. Wainstein and S. Kvatinsky, "A Lumped RF Model for Nanoscale Memristive Devices and Non-Volatile Single-Pole Double-Throw Switches," *IEEE Transactions on Nanotechnology*, p. 1, 2018.

[115] R. E. Pino, J. W. Bohl, N. McDonald, B. Wysocki, P. Rozwood, K. A. Campbell, A. Oblea, and A. Timilsina, "Compact method for modeling and simulation of memristor devices: Ion conductor chalcogenide-based memristor devices," in *IEEE/ACM International Symposium on Nanoscale Architectures*, 2010, Conference Proceedings, pp. 1–4.

[116] R. Berdan, C. Lim, A. Khiat, C. Papavassiliou, and T. Prodromakis, "A Memristor SPICE Model Accounting for Volatile Characteristics of Practical ReRAM," *IEEE Electron Device Letters*, vol. 35, no. 1, pp. 135–137, Jan 2014.

[117] A. G. Radwan and M. E. Fouda, *On the mathematical modeling of memristor, memcapacitor, and meminductor*. Springer, 2015, vol. 26.

[118] C. H. Bennett, J.-E. Lorival, F. Marc, T. Cabaret, B. Jousselme, V. Derycke, J.-O. Klein, and C. Maneux, "Multiscaled simulation methodology for neuro-inspired circuits demonstrated with an organic memristor," *IEEE Transactions on Multi-Scale Computing Systems*, 2017.

[119] A. G. Radwan, A. Taher Azar, S. Vaidyanathan, J. M. Munoz-Pacheco, and A. Ouannas, "Fractional-order and memristive nonlinear systems: Advances and applications," *Complexity*, vol. 2017, 2017.

[120] I. Messaris, A. Serb, S. Stathopoulos, A. Khiat, S. Nikolaidis, and T. Prodromakis, "A Data-Driven Verilog-A ReRAM Model," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 12, pp. 3151–3162, 2018.

[121] Z. Guo, G. Si, X. Xu, K. Qu, and S. Li, "Generalized modeling and character analyzing of composite fractional-order memristors in series connection," *Nonlinear Dynamics*, pp. 1–15, 2018.

[122] J. Singh and B. Raj, "Temperature dependent analytical modeling and simulations of nanoscale memristor," *International Journal of Engineering Science and Technology*, vol. 21, no. 5, pp. 862–868, 2018.

[123] R. Berdan, T. Prodromakis, and C. Toumazou, "High precision analogue memristor state tuning," *Electronics Letters*, vol. 48, no. 18, pp. 1105–1107, 2012.

[124] O. A. Olumodeji and M. Gottardi, "A pulse-based memristor programming circuit," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017, pp. 1–4.

[125] I. Vourkas, J. Gomez, A. A, N. Vasileiadis, G. C. Sirakoulis, and A. Rubio, "Exploring the voltage divider approach for accurate memristor state tuning," in *Latin American Symposium on Circuits & Systems (LASCAS)*, 2017, pp. 1–4.

[126] Q. Li, A. Serb, T. Prodromakis, and H. Xu, "A Memristor SPICE Model Accounting for Synaptic Activity Dependence," *PLOS ONE*, vol. 10, no. 3, p. e0120506, Mar 2015.

[127] S. Cortese, M. Trapatseli, A. Khiat, and T. Prodromakis, "A TiO2-based volatile threshold switching selector device with 10^7 non linearity and sub 100 pa off current," in *International Symposium on VLSI Technology, Systems and Application (VLSI-TSA)*, 2016, pp. 1–2.

[128] W. Sun, B. Gao, M. Chi, Q. Xia, J. J. Yang, H. Qian, and H. Wu, "Understanding memristive switching via in situ characterization and device modeling," 2019.

[129] N. Onofrio, D. Guzman, and A. Strachan, "Atomic origin of ultrafast resistance switching in nanoscale electrometallization cells," *Nature Materials*, vol. 14, no. 4, pp. 440–446, Apr 2015.

[130] R. Berdan, T. Prodromakis, A. Khiat, I. Salaoru, C. Toumazou, F. Perez-Diaz, and E. Vasilaki, "Temporal processing with volatile memristors," in *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*. IEEE, May 2013, Conference Proceedings, pp. 425–428.

[131] Z. Wang, S. Joshi, S. E. Savel'ev, H. Jiang, R. Midya, P. Lin, M. Hu, N. Ge, J. P. Strachan, Z. Li, Q. Wu, M. Barnell, G.-L. Li, H. L. Xin, R. S. Williams, Q. Xia, and J. J. Yang, "Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing," *Nature Materials*, vol. 16, no. 1, pp. 101–108, Jan 2017.

[132] B. Razavi, "The StrongARM Latch [A Circuit for All Seasons]," *IEEE Solid-State Circuits Magazine*, vol. 7, no. 2, pp. 12–17, 2015.

[133] J. S. Tandon, S. Komatsu, T. J. Yamaguchi, and K. Asada, "A comparative study of body biased time-to-digital converters based on stochastic arbiters and stochastic comparators," in *2016 14th IEEE International New Circuits and Systems Conference (NEWCAS)*. IEEE, Jun 2016, pp. 1–4.

[134] C. Seitz, "System timing," in *Introduction to VLSI Systems*, C. Mean and L. Conway, Eds. Addison-Wesley, 1979, book section 7.

[135] A. Soltan and A. M. Soliman, "A CMOS differential difference operational mirrored amplifier," *AEU-International Journal of Electronics and Communications*, vol. 63, no. 9, pp. 793–800, 2009.

[136] S. Sarkar and S. Banerjee, "500 MHz differential latched current comparator for calibration of current steering DAC," in *IEEE Symposium on Students' Technology (TechSym)*, 2014, pp. 309–312.

[137] Z. Lak and N. Nicolici, "On Using On-Chip Clock Tuning Elements to Address Delay Degradation Due to Circuit Aging," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 12, pp. 1845–1856, Dec 2012.

[138] C. Huang, F. Liu, Q. Wang, and Z. Huo, "A Small Ripple Program Voltage Generator Without High-Voltage Regulator for 3D NAND Flash," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 6, pp. 1049–1053, Jun 2020.

[139] Z. Biolek, D. Biolek, and V. Biolkova, "SPICE Model of Memristor with Nonlinear Dopant Drift," *Radioengineering*, vol. 18, no. 2, p. 0, 2009.

[140] V. A. Slipko and Y. V. Pershin, "Importance of the window function choice for the predictive modelling of memristors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2019.

[141] E. J. Merced-Grafals, N. Davila, N. Ge, R. S. Williams, and J. P. Strachan, "Repeatable, accurate, and high speed multi-level programming of memristor 1T1R arrays for power efficient analog computing applications," *Nanotechnology*, vol. 27, no. 36, p. 365202, 2016.

[142] V. Constantoudis, G. Papavieros, P. Karakolis, A. Khiat, T. Prodromakis, and P. Dimitrakis, "Impact of Line Edge Roughness on ReRAM Uniformity and Scaling," *Materials*, vol. 12, no. 23, p. 3972, Nov 2019.

[143] I. Vourkas, A. Batsos, and G. C. Sirakoulis, "SPICE modeling of nonlinear memristive behavior," *International Journal of Circuit Theory and Applications*, vol. 43, no. 5, pp. 553–565, 2015.

[144] Y. N. Joglekar and S. J. Wolf, "The elusive memristor: properties of basic electrical circuits," *European Journal of Physics*, vol. 30, no. 4, p. 661, 2009.

[145] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, "Digital Integrated Circuits (2nd Edition)," *Book*, 2003.

[146] Y. Wang and M. Zwolinski, "Analytical transient response and propagation delay model for nanoscale CMOS inverter," in *2009 IEEE International Symposium on Circuits and Systems*. IEEE, May 2009, pp. 2998–3001.

[147] Z. Huang, A. Kurokawa, M. Hashimoto, T. Sato, M. Jiang, and Y. Inoue, "Modeling the Overshooting Effect for CMOS Inverter Delay Analysis in Nanometer Technologies," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 2, pp. 250–260, Feb 2010.

[148] A. M. Hassan, H. A. Fahmy, and N. H. Rafat, "Enhanced model of conductive filament-based memristor via including trapezoidal electron tunneling barrier effect," *IEEE Transactions on Nanotechnology*, vol. 15, no. 3, pp. 484–491, 2016.

[149] G. A. Illarionov, S. M. Morozova, V. V. Chrishtop, M.-A. Einarsrud, and M. I. Morozov, "Memristive TiO2: Synthesis, Technologies, and Applications," *Frontiers in Chemistry*, vol. 8, Oct 2020.

[150] E. Pérez, C. Wenger, A. Grossi, C. Zambelli, P. Olivo, and R. Roelofs, "Impact of temperature on conduction mechanisms and switching parameters in HfO 2 -based 1T-1R resistive random access memories devices," *Journal of Vacuum Science & Technology B, Nanotechnology and Microelectronics: Materials, Processing, Measurement, and Phenomena*, vol. 35, no. 1, p. 01A103, Jan 2017.

[151] S. Kim, S.-J. Kim, K. M. Kim, S. R. Lee, M. Chang, E. Cho, Y.-B. Kim, C. J. Kim, U. In Chung, and I.-K. Yoo, "Physical electro-thermal model of resistive switching in bi-layered resistance-change memory," *Scientific Reports*, vol. 3, p. 1680, 2013.

[152] M. Mikami and K. Ozaki, "Thermoelectric properties of nitrogen-doped TiO_{2-x} compounds," *Journal of Physics: Conference Series*, vol. 379, no. 1, p. 012006, 2012.

[153] G. A. Patterson, F. Sangiuliano Jimka, P. I. Fierens, and D. F. Grosz, "Memristors under the influence of noise and temperature," *physica status solidi c*, vol. 12, no. 1-2, pp. 187–191, 2015.

[154] D. Mardare, C. Baban, R. Gavrila, M. Modreanu, and G. I. Rusu, "On the structure, morphology and electrical conductivities of titanium oxide thin films," in *Surface Science*, 2002.

[155] B. Cook, "Electrical Switching Studies of Chalcogenide-Based Ion-Conducting Variable Resistance Devices," *Boise State University Theses and Dissertations*, May 2011.

[156] P. Li, Y. Zhang, Z. Chen, P. Gao, T. Wu, and L.-M. Wang, "Relaxation dynamics in the strong chalcogenide glass-former of $Ge_{22}Se_{78}$," *Scientific Reports*, vol. 7, no. 1, p. 40547, Feb 2017. [Online]. Available: http://www.nature.com/articles/srep40547

[157] S. Banerjee and D. Chakravorty, "Electrical resistivity of silver-silica nanocomposites," *Journal of Applied Physics*, vol. 85, no. 7, pp. 3623–3625, Apr 1999.

[158] W. Woods, M. M. A. Taha, S. J. Dat Tran, J. Burger, and C. Teuscher, "Memristor panic - A survey of different device models in crossbar architectures," in *Proceedings of the 2015 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCHÂŽ15)*. IEEE, Jul 2015, Conference Proceedings, pp. 106–111.

[159] M. Hu, C. E. Graves, C. Li, Y. Li, N. Ge, E. Montgomery, N. Davila, H. Jiang, R. S. Williams, J. J. Yang, Q. Xia, and J. P. Strachan, "Memristor-based analog computation and neural network classification with a dot product engine," *Advanced Materials*, vol. 30, no. 9, p. 1705914, 2018.

[160] M. Escudero-Lopez, E. Amat, A. Rubio, and P. Pouyan, "An experience with chalcogenide memristors, and implications on memory and computer applications," in *2016 Conference on Design of Circuits and Integrated Systems (DCIS)*. IEEE, Nov 2016, Conference Proceedings, pp. 1–6.

[161] A. Serb, W. Redman-White, C. Papavassiliou, and T. Prodromakis, "Practical Determination of Individual Element Resistive States in Selectorless RRAM Arrays," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 6, pp. 827–835, Jun 2016.

[162] B. Haaheim and T. G. Constandinou, "A sub-1μW, 16kHz current-mode SAR-ADC for single-neuron spike recording," in *2012 IEEE International Symposium on Circuits and Systems*. IEEE, May 2012, Conference Proceedings, pp. 2957–2960.

[163] A. Elkafrawy, J. Anders, and M. Ortmanns, "Design and validation of a 10-bit current mode SAR ADC with 58.4 dB SFDR at 50 MS/s in 90 nm CMOS," *Analog Integrated Circuits and Signal Processing*, vol. 89, no. 2, pp. 283–295, 2016.

[164] J. Carreira, "A two-step flash ADC for digital CMOS technology," in *Second International Conference on Advanced A-D and D-A Conversion Techniques and their Applications*, vol. 1994. IEEE, 1994, Conference Proceedings, pp. 48–51.

[165] J. Zhou, D. J. Kinniment, C. E. Dike, G. Russell, and A. V. Yakovlev, "On-chip measurement of deep metastability in synchronizers," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 550–557, 2008.

[166] G. Fuchs, M. Függer, and A. Steininger, "On the threat of metastability in an asynchronous fault-tolerant clock generation scheme," in *IEEE Symposium on Asynchronous Circuits and Systems*, 2009, pp. 127–136.

[167] B. Cheng, A. Emboras, Y. Salamin, F. Ducry, P. Ma, Y. Fedoryshyn, S. Andermatt, M. Luisier, and J. Leuthold, "Ultra compact electrochemical metallization cells offering reproducible atomic scale memristive switching," *Communications Physics*, vol. 2, no. 1, p. 28, Dec 2019.

[168] P. Nuzzo, C. Nani, C. Armiento, A. Sangiovanni-Vincentelli, J. Craninckx, and G. V. d. Plas, "A 6-bit 50-MS/s threshold configuring SAR ADC in 90-nm digital cmos," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 1, pp. 80–92, 2012.

[169] M. E. Lee, W. J. Dally, and P. Chiang, "Low-power area-efficient high-speed I/O circuit techniques," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1591–1599, 2000.

[170] S. J. Kim, D. Kim, and M. Seok, "Comparative study and optimization of synchronous and asynchronous comparators at near-threshold voltages," in *IEEE/ACM International Symposium on Low Power Electronics and Design*, 2017, pp. 1–6.

[171] X. Zhong, A. Bermak, and C. Y. Tsui, "A low-offset dynamic comparator with area-efficient and low-power offset cancellation," in *IFIP/IEEE International Conference on VLSI-SoC*, 2017, pp. 1–6.

[172] X. Xin, J. Cai, R. Xie, and P. Wang, "Ultra-low power comparator with dynamic offset cancellation for SAR ADC," *Electronics Letters*, vol. 53, no. 24, pp. 1572–1574, 2017.

[173] M. Nasrollahpour and S. Hamedi-Hagh, "High speed, low offset, low power differential comparator with constant common mode voltage," in *2017 IEEE International Conference on ASIC*, 2017, pp. 871–874.

[174] J. E. M. Solis, M. G. Navarro, I. Mejia, R. Z. G. Lozano, F. L. Rojas, J. Ocampo-Hidalgo, and H. B. d. Toro, "Low input resistance CMOS current comparator based on the FVF for low-power applications," *Canadian Journal of Electrical and Computer Eng.*, vol. 39, no. 2, pp. 127–131, 2016.

[175] P. Suriyavejwongs, E. Leelarasmee, W. Pora, and S. Tontisirin, "Inverting amplifier based ultra low power low offset current comparator," in *International Electrical Engineering Congress (iEECON)*, 2017, pp. 1–4.

[176] T. Jankatkit and V. Kasemsuwan, "Low-voltage current-mode preamplifier based latch comparator," in *International Conference on Electrical Engineering/Electronics, Computer, Telecom. and Information Tech.*, 2016, pp. 1–4.

[177] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran," in *IEEE International Symposium on Circuits and Systems (ISCAS 85)*. IEEE Press, Piscataway, N.J., 1985, pp. 677–692.

[178] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits," in *IEEE International Symposium on Circuits and Systems*, 1989.

[179] C. Albrecht, "IWLS 2005 Benchmarks," in *2005 International Workshop on Logic Synthesis*, Jun 2005.