Towards Designing Thermally-Aware Memristance Decoder

Thanasin Bunnam, *Student Member, IEEE,* Ahmed Soltan, *Member, IEEE,* Danil Sokolov, *Member, IEEE,* Oleg Maevsky, Alex Yakovlev, *Fellow, IEEE*

***Abstract*—Memristors are intensively proposed in many applications, such as biosensors and machine learning. Regarding their analog characteristics, memristance decoder is, therefore, an essential part for every memristor-based system. As memristor is a temperature sensitive device, this work proposes a memristance decoder circuit with self-temperature calibration. Its main building block is a comparator which is based on current mode circuit to achieve high performance at low power. The design provides configurable precision based on the available energy and supports both synchronous and asynchronous schemes. Moreover, the VTEAM model is modified to include the temperature effect on the memristance in the analysis. The simulation results, based on UMC 65nm low-leakage CMOS technology, show the following comparator's characteristics: 1.70% maximum offset, $2.91ns$ worst case latency, $343MHz$ maximum frequency and $48.79fJ$ maximum energy per comparison. Monte Carlo simulation shows the metastable state in determining the memristor value. This can be solved by extending the clock period or applying a metastability resolver. The proposed memristor model reveals that memristance at high resistive state degrades quadratically with the rise of the temperature and at $85°C$ nearly reaches the memristance of low resistive state. The possibility of decoding error due to the temperature effect is demonstrated via simulations.**

***Keywords*—memristor, thermal effect, resistance comparator, memristance decoder, metastability**

I. Introduction

The memristor response has been defined for the first time by Chua in 1971 [1] while the first implementation based on TiO_2 is presented in 2008 by [2]. The main advantage of the memristor is that it can maintain its internal resistance, called memristance, even in the absence of power. Hence, it is considered as a passive memory element for the systems with unstable energy sources [3]. For internet of things (IoT) applications, memristors are also utilised in the pre-processing stage for sensing purposes, such as image processing [4]. This allows to save power by transmitting less information to the cloud. Being compatible with CMOS process, exhibiting high endurance and long data retention, memristors are also applied in the following research frontiers: multi-bit memory [5], in-memory computing [6], [7], arithmetic and logic operations [8]–[10], neural network and neuromorphic computing [11]–[14], time adaptive circuit [15], [16], biosensor [17]–[20] and the mixed use of temperature sensor and memory [21]. Although the memristor is intensively used in many applications, neither of the previously reported works discussed the memristor decoder, which is the key component for the memristance reading process.

A decoder needs a comparator for decoding the analog value, such as the memristance. The comparator indicates if the memristor's current/voltage, which reflects the memristance, is greater or less than the reference one. Therefore, the memristance decoding process is iterative comparison against several reference values until the correct data is found. To improve the resolution of the memristance reading, multiple high-precision comparators are required. However, both increasing the number of comparator and improving the accuracy cause longer latency and higher power consumption. This is the critical issue for the edge devices in IoT systems because they are typically operated by the unstable and limited power sources, such as the energy harvesters and batteries. Furthermore, the electronic design is moving towards power adaptive scheme in which the system can trade its accuracy for power savings in order to survive under any energy conditions [3], [22]. For this reason, designing the low-power memristance decoder with the comparator capable of scaling its resolution accuracy based on the available energy budget, is one of our design challenges.

Temperature fluctuations influence the performance of all electronic devices in the system, including the decoder and the memristor. This especially impacts on the edge processing IoT devices operating on remote sites, due to the changes of environmental conditions. In addition, the devices can heat up due to high computation activity, such as image processing and deep learning. Traditionally, temperature sensors are employed to monitor the temperature and adjust the system performance accordingly. This approach relies on models of the system component performance as a function of temperature. However, most of the reported memristor models do not include the impact of the temperature [2], [23]–[31]. There are few temperature-embedded mathematical models presented in [32], [33], however they did not provide the simulator-based model, such as Verilog-A. Furthermore, implementing multiple complex equations in Verilog-A can degrade the simulator efficiency [26] and cause the convergence problem [34]. The model in [21] is proposed for building the memristor crossbar that can sense the circuit temperature and keep the data at the same time. Nevertheless, its linear I-V relationship does not fit the practical devices, which are highly nonlinear. Overall, the accurate temperature effect must be embedded in the memristor model, preferably with a negligible simulation overhead. This is necessary for studying the memristance

T. Bunnam, A. Soltan, D. Sokolov, and A.Yakovlev are with the School of Engineering, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K. (e-mail: t.bunnam2@newcastle.ac.uk; ahmed.abd-el-aal@newcastle.ac.uk; danil.sokolov@newcastle.ac.uk; alex.yakovlev@newcastle.ac.uk).

O. Maevsky is with Nanodevices LTD, Russia (e-mail: oleg-maevsky@yandex.ru).

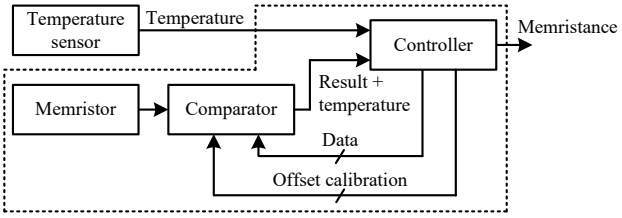

Fig. 1. Block diagram of the proposed decoder. The controller programs the reference of the comparator via the data signals and requests for the comparison. This process iterates until the comparator's output toggles. Next, the controller compensates the output using the temperature reading from the sensor. This work focuses on the memristor model with thermal effect, the comparator and the controller designs (these components are outlined by the dashed line).

decoding under the temperature variation and investigating the compensation techniques.

In this work, a procedure for embedding the temperature effect into the memristor models is presented. Our analysis is based on the VTEAM model, which is implemented in Verilog-A [27], and can be used with the SPICE model presented in [35]. This enables the study of temperature effect and its compensation techniques, which is paramount for all analog applications of the memristor. This also impacts the memristor-based biosensors as some of them require a certain temperature configuration to operate properly [19], [29]. The modified model is validated for TiO_2 by deriving the model parameters from the practical device measurements presented in [32], [36] and can be applied for any kind of memristor.

This paper also proposes a novel decoder design for analog memristor applications. It is used for demonstrating the impact of temperature on the memristance reading process. The decoder uses a comparator which is based on current mode circuits and thus inherits their intrinsic advantages over the voltage one, such as like low power, wide bandwidth and less susceptible to power supply fluctuation [37], [38]. The comparator is designed to provide the resolution scaling based on the available power budget and supports both synchronous and asynchronous schemes. In addition, this design is resistor-free to save the chip area, which is a critical issue in edge device design [39], and to avoid any resistor induced variations.

The simplified block diagram of the proposed decoder is shown in Fig. 1. The decoder consists of a comparator and a controller to read and decode the memristor value. Furthermore, the controller calibrates the comparator offset which comes from the process variation. It also calibrates the read value using the temperature data from the sensor. To evaluate the function and performance, 2-bit decoder is revealed. It is able to be scaled to n-bit based on the required accuracy and the available power budget.

To summarise, the main contributions of this work are as follows:

- A modified model of memristance with temperature effect, and
- A novel energy-scalable memristance decoder design with temperature awareness.

This paper is organised as follows. The thermal effect on the memristance is analysed and embedded into the VTEAM model in Section II. Section III explains the resistance decoder design and the metastability issue. Section IV shows the performance evaluation of the decoder and the impact of the temperature on the memristance reading. Because there is no memristance decoder in the literature, the performance of the comparator, which dominates the overall performance, is compared against the related works instead in Section V. Finally, the conclusion is provided in Section VI.

II. Thermal Memristance Model

A. Model Analysis

The temperature effect on the memristance will be analysed and embedded in the VTEAM model because it precisely estimates all reported physical devices behaviours, such as linear ion drift [2], [21], nonlinear ion drift [23] and Simmons tunnel barrier [24], yet exhibiting better computation efficiency [27]. The memristance R_m is expressed in the model as in (1):

$$R_m = R_{on} + (R_{off} - R_{on})\frac{w(t)}{D} \tag{1}$$

$$\frac{dw(t)}{dt} = \begin{cases} k_{off}(\frac{v(t)}{v_{off}} - 1)^{\alpha_{off}} f_{off}(w), & 0 < v_{off} < v \\ 0, & v_{on} < v < v_{off} \\ k_{on}(\frac{v(t)}{v_{on}} - 1)^{\alpha_{on}} f_{on}(w), & v < v_{on} < 0 \end{cases} \tag{2}$$

where D is the distance between the lower and higher bounds (w_{on} and w_{off}) of state variable $w(t)$, R_{on} and R_{off} are the memristances corresponding to those bounds. From (2), $w(t)$ depends on the amplitude, polarity and duration of the applied voltage $v(t)$. Note that v_{on} and v_{off} are the threshold voltages, $f_{on}(w)$ and $f_{off}(w)$ are the window functions and the remaining variables are the fitting parameters.

This work focuses on the valence change memory (VCM) based memristors [40] because it provides a quantum resistance which is useful for multi-level and neuromorphic computing applications. From Poisson-Boltzmann equation, the temperature effect on the memristance is minimal when the oxygen vacancies are spread through the device [32], [41]. Thus, the effect on R_{on} is negligible compared to the one on R_{off}. For this reason, the following analysis studies only embedding the thermal effect on R_{off} into VTEAM model. Note that the thermal effect on R_{on} can be easily embedded in the model by following the same procedure.

The basic resistance model is given as follows:

$$R = \rho \times \frac{L}{A} \tag{3}$$

where R, ρ, L and A are resistance, resistivity, length (width in this case) and cross section area of the memristor respectively. From (3), the resistance is proportional to the resistivity of the material which is temperature dependent as given in (4):

$$\rho = \rho_0 \exp^{-\alpha T} \tag{4}$$

where ρ_0 is the resistivity at room temperature ($25°C$) and α is the temperature coefficient. This equation shows that the resistivity decreases with increasing temperature, which agrees with the reported property in [42]. Also, it can be applied to the

TABLE I
CALCULATED OFF RESISTANCE OF TITANIUM DIOXIDE MEMRISTOR

Width L (nm)	R_{off} from [32] $(M\Omega)$	Calculated R_{off} $(M\Omega)$	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 II
THERMAL EFFECT ON RESISTIVITY OF TITANIUM DIOXIDE MEMRISTOR

Temperature $(°C)$	R_{off}/R_{on} ratio from [32]	Calculated R_{off} $(k\Omega)$	Calculated resistivity (Ωm)
25	15.0×10^3	1,950.00	158.00×10^2
65	4.0×10^3	520.00	42.10×10^2
85	1.0×10^3	105.00	10.50×10^2
125	0.5×10^3	0.53	5.27×10^2

other materials such as tantalum [43] and manganite [44]. For simpler computation, the resistivity can be approximated to the second order polynomial given in (5) where, C_0, C_1, and C_2 are material dependent constants. As the impact of temperature on R_{on} is negligible, equation (5) is used to represent the temperature effect on R_{off} in (1). Hence, the VTEAM model is modified as in (6).

$$\rho = C_2T^2 + C_1T + C_0 \tag{5}$$

$$\begin{aligned} R_m =& R_{on}(1 - \frac{w(t)}{D}) + \\ & (C_2T^2 + C_1T + C_0)\frac{L}{A}\cdot\frac{w(t)}{D} \end{aligned} \tag{6}$$

B. Model Verification

To verify the model, firstly, the equation (3) is used to calculate the resistivity of TiO_2 $(16,000\Omega m)$ using the measured data from [36] ($R_{off} = 4M\Omega, A = 100\mu m^2$ and $L = 25nm$). Then, the R_{off} at different device's widths are estimated using the calculated resistivity compared to the experimental data from [32] as listed in Table I. From the table, the maximum error of 1.25% is at $L = 20nm$ which shows the accuracy of the proposed analysis.

TABLE III
CALCULATED VALUE FOR THE CONSTANTS OF (6)

C_2	C_1	C_0
2.3034	-498.4	26,826

TABLE IV
VTEAM MODEL PARAMETERS (PARTIALLY FROM [45])

Parameter	Value	Unit
α_{off}	4	–
α_{on}	4	–
v_{off}	0.3	V
v_{on}	-1.5	V
R_{on}	1,000	Ω
k_{off}	0.091	m/s
k_{on}	-216.2	m/s
w_{off}	3×10^{-9}	nm
w_{on}	0	nm
L	w_{off}	nm
A	12×12	μm^2

Secondly, the R_{off} and the resistivity at different temperatures are estimated using the extracted values ($R_{on} = 130\Omega, A = 9\times9\mu m^2, L = 10nm$ and $, R_{off}/R_{on}$ ratios) from [32] as listed in Table II. Finally, the constants of (6) are calculated numerically using the obtained resistivity and basic fitting algorithm. The determined constants for TiO_2 are tabulated in Table III. Although the obtained parameters fit the data from [32], the accuracy comparison with the other works cannot be done because they do not provide enough information. For example, the work in [42] reports only the trend of the temperature and the resistivity, but it does not provide the exact values. By using same procedure defined here, the parameters of (6) can be recalculated for memristors with different materials such as ZnO and Ta_2O_5.

Embedding the modified model of (6) with the VTEAM model of (2) enables thermal analysis using circuit simulator and computing software such as Cadence Spectre and MATLAB. The thermal analysis using MATLAB for our model is illustrated in Fig. 3. The memristance tuning runs for up and down directions with an excitation time of $10ns$. Both sub-figures show the memristance decreases significantly when the temperature raises. Furthermore, it is close to

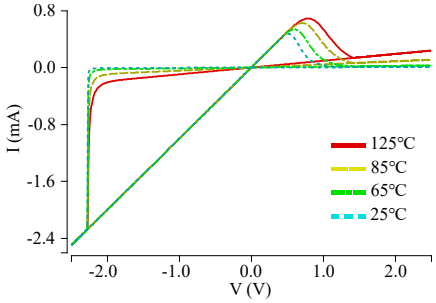

Fig. 2. I-V characteristic of the proposed model at different temperatures. At the higher temperature, the memristor at OFF state delivers the higher current due to the lower R_{off}. However, the current is temperature independent when the memristor operates at ON state because the thermal effect is neglect.

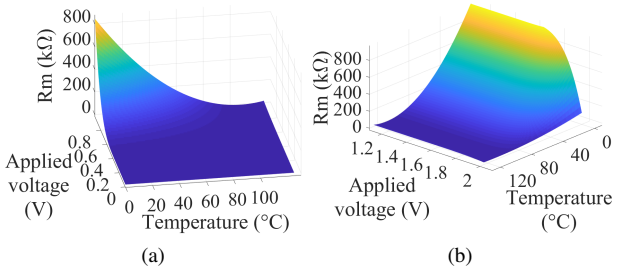

Fig. 3. The relationship between temperature, applied voltage and memristance with the excitation time of $10ns$. (a) Tune up. (b) Tune down. Our model reveals the temperature impact is higher when the memristance increases. The memristance is close to R_{on} once the temperature reaches $85°C$.

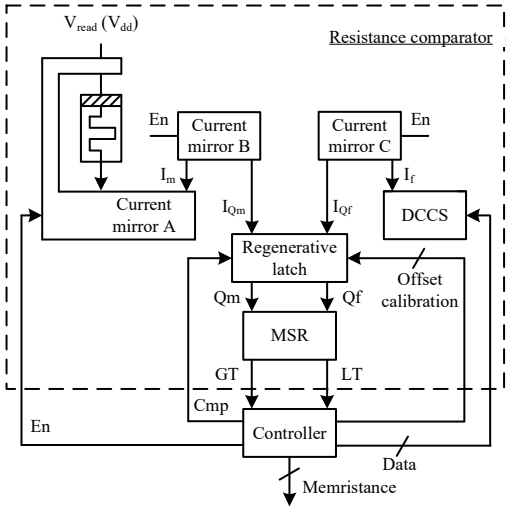

Fig. 4. Block diagram of the proposed decoder. The currents generated by the memristor (I_m) and the reference source (I_f) are duplicated by the current mirrors B and C (I_{Qm}, I_{Qf}). They are compared by the regenerative latch which results at Q_m and Q_f. The metastability caused by the latch is filtered by the metastability resolver (MSR). The controller determines the MSR's outputs, adjusts the data (reference source) and repeat the process until the MSR's outputs toggle (the uncompensated memristance is found). The signals En and Cmp are used to control the comparison process described by Fig. 5.

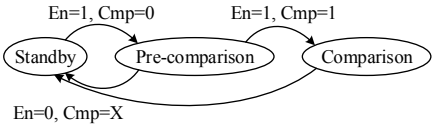

Fig. 5. State diagram of the memristance comparison process. En is used to switch from standby to pre-comparison state while Cmp is used to switch the state from pre-comparison to comparison. The state returns to standby and ready for the next iteration once En is 0.

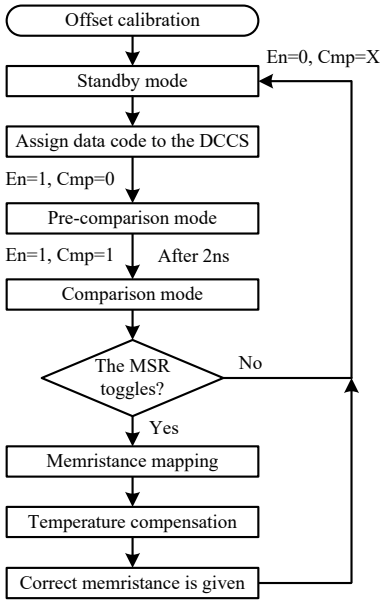

Fig. 6. Flowchart of the memristance decoding process. The decoder repeats the memristance comparison until the latch's outputs change (the uncompensated memristance is found). Then, the temperature compensation takes place to yield the correct value.

TABLE V
OPERATION MODES

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

TABLE VI
RESISTANCE COMPARATOR RESULTS

Result	Signal	
	Q_m, Q_f	GT, LT
Standby, pre-comp.	0, 0	0, 0
$I_m > I_f$	1, 0	1, 0
$I_m < I_f$	0, 1	0, 1

R_{on} when the temperature reaches approximately $85°C$. This means R_{off} is discarded at high temperature. Moreover, the memristance changes non-linearly depending on the applied voltage as described by the original VTEAM model. Note that the VTEAM parameters are retrieved from [45] and are summarised in Table IV. They fit the physical devices reported in [11]. In addition, both L and w_{off} are the same parameters which are the device's width. The cross section area A is selected such that R_{off} ($329k\Omega$) becomes close to the original parameter ($300k\Omega$).

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

III. MEMRISTANCE DECODER

A. Operation Principles and Design

The decoder operation is based on detecting and digitally decoding the current flowing through the memristor. Then, the decoded data is mapped to the actual memristance via a look-up table inside the controller. Finally, the measured value is compensated for the temperature variation to yield the correct one. Temperature compensation process is fulfilled using the temperature model described in the previous section.

The block diagram of the proposed decoder is illustrated in Fig. 4. It consists of a controller and a resistance comparator. Similar to the successive approximation (SAR) ADC [46], [47], the controller is used to run the search algorithm (e.g. binary search), memristance mapping and temperature compensation. It uses En and Cmp to control the comparator operation which is described by the state diagram in Fig. 5. 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.

The resistance comparator is similar to [48]. 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 both currents and indicates the larger/smaller currents at Q_m and Q_f. The metastability resolver (MSR) is attached to both latch's outputs to filter out the metastable state [49].

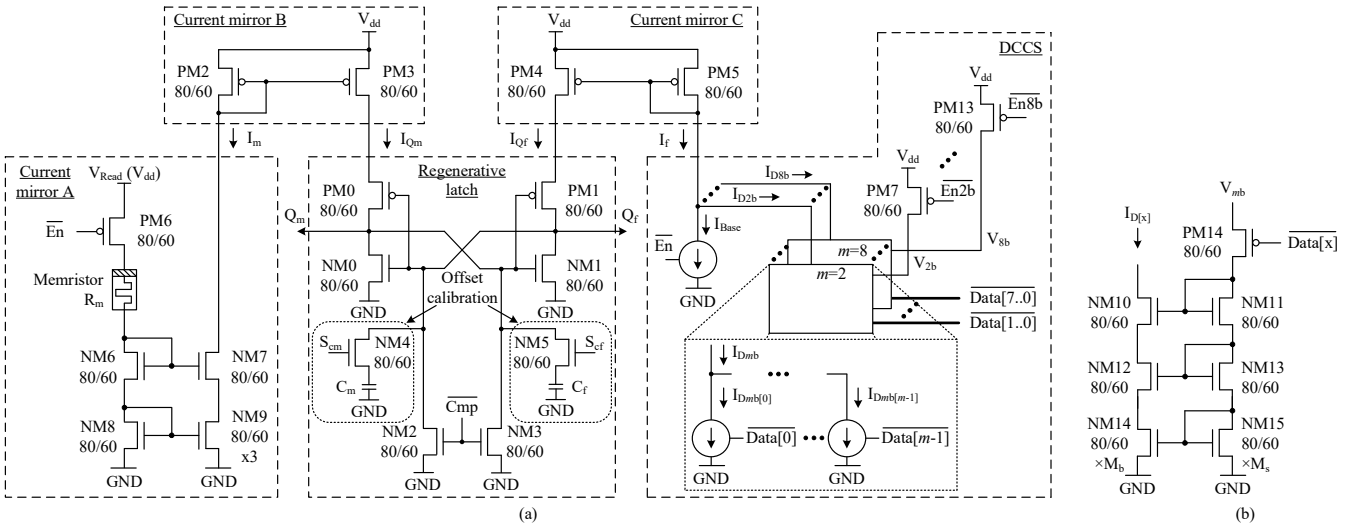

Fig. 7. (a) The resistance comparator circuit. The memristor current is generated by the current mirror A (I_m) while the reference current is produced by the DCCS (I_f). The DCCS contains multiple banks of current sources to support multiple resolutions which can be selected by $Enmb$ (En in Fig. 4- 6). 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. (b) The circuit for the current sources in DCCS. $I_{D[x]}$ is controlled by the number of NM12 and NM13 (M_b and M_s). The numbers of transistors for the base and data bits are listed in the table.

According to the state diagram, flowchart and circuit implementation in Fig. 5-7, the decoding process starts at the standby mode where En is set as 0 by the controller. This discharges the result nodes Q_m and Q_f via NM2-3 and disables the DCCS and all current mirrors to minimise the power consumption. Then, the decoded data is initialised to set up the reference current (DCCS). Subsequently, the comparator steps into the pre-comparison mode by toggling En to 1 for enabling all current mirrors and the DCCS. As a result, I_{Qm} and I_{Qf} flow through the latch towards ground. In order to ensure the stability of both currents, this state has to be held for a certain amount of time ($2ns$).

The comparison starts by switching Cmp to 1 for disabling NM2-3. As a result, I_{Qm} and I_{Qf} flow against each other at Q_m and Q_f. This causes the metastability at both nodes before the outputs indicating the larger/smaller currents are valid. Therefore, the MSR is applied to eliminate such problem and the results are given as GT and LT as summarised in Table VI. 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 the data is found. Then, the controller maps the decoded data to the memristance using a look-up table. Finally, it adjusts the measured memristance regarding the current temperature, which is modelled in the previous section, to yield the correct value. The relation between the comparator's state and the control signals is summarised in Table V. Also, Table VI summarises the relation 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 the resolution scalability, the reference current (I_f) is configurable as it sums the currents from the base and the DCCS (Fig. 7a). The DCCS is implemented as multiple banks of current sources. Each bank contains a number of parallel current sources (Fig. 7b) with respect to the number of bits denoted by m. Also, each bank is enabled by the heading transistor which is controlled by $Enmb$. 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.

Regarding the circuit implementation, the transistors PM2/PM5 and PM3/PM4 have to be matched to avoid any current offsets. To minimise the offset caused by the mismatch between the current mirror A and the DCCS, both sub-circuits are isolated from the latch by the current mirrors B and C.

In summary, the proposed decoder provides a flexible choice of speed and power trade off as the sizes of current mirrors can be either increased for speed improvement or decreased for power saving. It offers the scalability in which the 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 does not use any resistors which helps to reduce the chip area and also makes it immune to resistivity variations. In case of the system with multiple memristors (e.g. crossbar), using a multiplexer can minimise the chip area as the devices can be read by a single decoder [21].

B. Data Range and Decoding

The decoder is designed for a specific range of memristance. It provides different precision based on the available energy and the required accuracy. The DCCS consists of different digitally controlled current mirrors as mentioned in the previous section. The resolution is calculated using the relation: $NumberOfBits = log_2(MemristanceRange/StepSize)$. Then, the current generated by each data boundary is observed and used to determine the transistor size of each current source (Fig. 7b). An example of 2-bit code for a memristance range of $134.257k\Omega - 324.801k\Omega$ with approximately $80k\Omega$

TABLE VII
DATA RANGE, LATENCY AND ENERGY AT $25°C$

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

step size is tabulated in Table VII. A safety gap must be allocated at each end to avoid the uncertainty of R_{on} and R_{off} due to the process variation. Furthermore, the gap at the low memristance side should be widened to ensures the saturation (where the low memristance causes insufficient voltage for memristor programming) does not occur [50]. Note that, the memristance in Table VII defines the upper bound of the range. For example, data 00 is defined between $237.506k\Omega - 324.801k\Omega$. In addition, the upper bound of 11 is higher than the specification ($80k\Omega$ gap) because the combination of $Data[0]$ and $Data[1]$ yields the higher current.

C. Metastability

The proposed comparator enters the metastable state during the decision making as revealed in Fig. 8 (Q_m and Q_f). Its magnitude is greater and its duration is longer when the memristance is close to the data boundary because I_{Qm} is nearly equal to I_{Qf}. The metastability can be caused by the device mismatch as illustrated in Fig. 9. The samples with long latency indicate the high impact of the metastable state. The metastability will cause a sampling error in synchronous circuit when the clock arrives the receiver, e.g. flip-flop, during the unstable state [51]. This issue is more significant in the case of asynchronous circuit because the successor stage can step to the incorrect sequence due to the metastable input [52].

A metastability resolver (MSR) is attached to Q_m and Q_f to filter out such the signals [49]. The circuit implementation

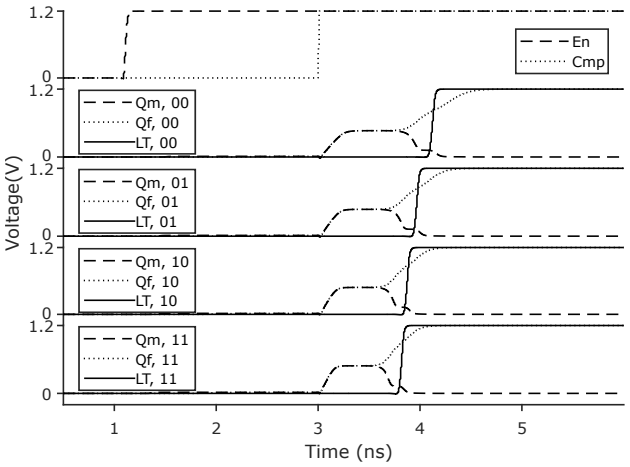

Fig. 8. Simulation results of the resistance comparator at $25°C$. Each simulation runs with the effective R_m in Table VII to show the maximum metastability (Q_m and Q_f) which is filtered by the MSR (LT).

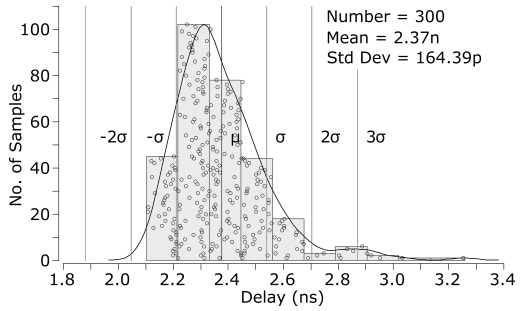

Fig. 9. Monte Carlo simulation reveals the latency distribution of the resistance comparator without the current mirror A and DCCS (300 samples). It shows the device mismatch causes the metastability which affects the decision time. The clock period must be greater than $2.54ns$ to cover the yield at 1σ.

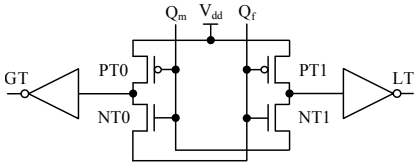

Fig. 10. The metastability resolver (MSR). PT0 and PT1 are ON and V_{dd} is connected to both inverters even when both inputs (Q_m, Q_f) enters the metastable state. Once the metastability vanishes, the complementary outputs are sent to both inverters.

is illustrated in Fig. 10. When the comparator is in the standby mode, logic 0s at Q_m and Q_f turn on PT0 and PT1 so that both inverter's inputs are 1s (GT and LT are 0s). These PMOSs are still ON during the metastable state. Once the logic becomes fully differential, either PT0/NT1 or PT1/NT0 are ON accordingly and give the result as listed in Table VI.

From the simulation result in Fig. 8, the memristance is set at the boundary to implement the worst case scenario where the metastability is maximised as I_{Qf} and I_{Qm} are nearly equal. It shows the metastable state is longer at lower data because the smaller currents are generated (higher memristance). However, such metastability is completely removed by the MSR (GT, LT). Hence, the inserted MSR prevents the failure in the rest of the system that is caused by this ambiguous state.

IV. SIMULATION RESULTS

There are three simulations to determine: offset, latency and energy, and thermal effect of our design. All circuits are implemented using UMC 65nm low-leakage CMOS technology. The VTEAM parameters, which are obtained from the practical device [45]), and our extracted parameters are listed in Table IV. All simulations are done using Cadence Spectre.

A. Offset

Although the DCCS has been tuned to deliver the same amount of currents as the expected memristances (boundaries) in Table VII, the difference in settle time between I_m and I_f, due to the difference in parasitic capacitance between the current mirror A and the DCCS, still causes the offset. To measure the offset, the comparison process is performed with the expected memristance in Table VII.

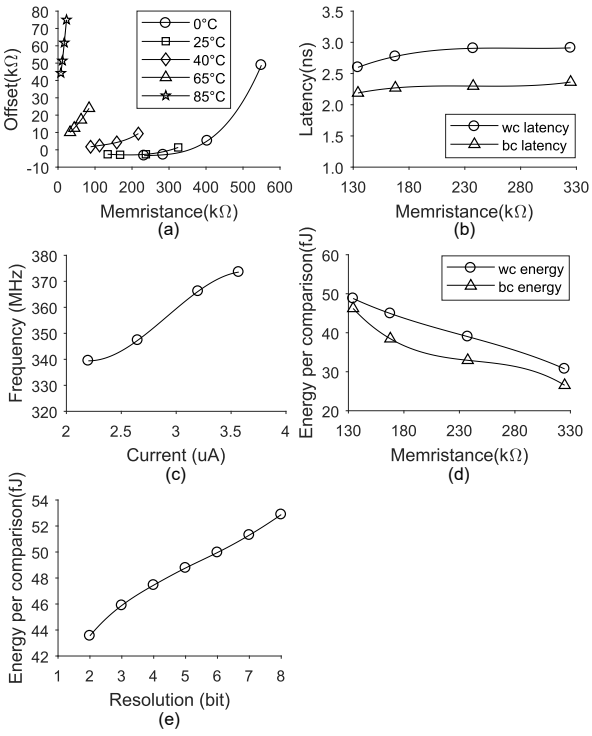

Fig. 11. Performances of the resistance comparator (a) Offsets of the resistance comparator at $25°C$. They are from subtracting the expected R_m by the effective one (Table VII). The maximum offset is 1.91% of the expected value. (b) The highest latency ($2.91ns$) is at 00 which has the highest memristance and thus lowest current. (e) The maximum frequency increases in the same way as the latch's input currents. (d) The worst case energy is higher than the best case one at every data because the metastability causes the longer comparison time. (e) The maximum energy per comparison at the largest data is higher when the resolution increases due to the lower memristance.

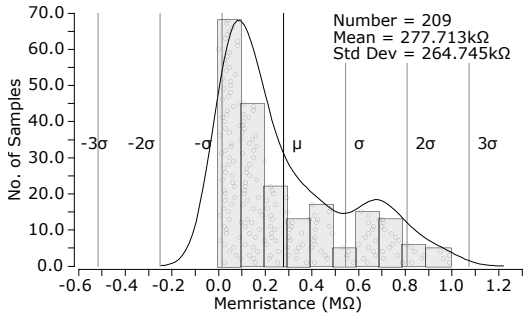

Fig. 12. The detected memristance when the DCCS is set as 00. Most of the samples fall below the desired range, which is 00. This can be solved by using the offset calibration techniques such as programmable capacitor array.

Then, the memristance is swept until the effective memristance which actually causes the change of the latch's outputs is found. The difference between two values is considered as the offset. Fig. 11(a) shows the offset is positive when the data is at 00 and negative otherwise. The maximum absolute offset is $2.853k\Omega$ (data 10) which is only 1.70% of the expected memristance ($167.761k\Omega$).

The same procedure is repeated to determine the offset at different temperatures as illustrated in Fig. 11(a). It shows the absolute offset is less than $10k\Omega$ when the memristor works between $25 - 40°C$. Otherwise, the offset will increase espe-

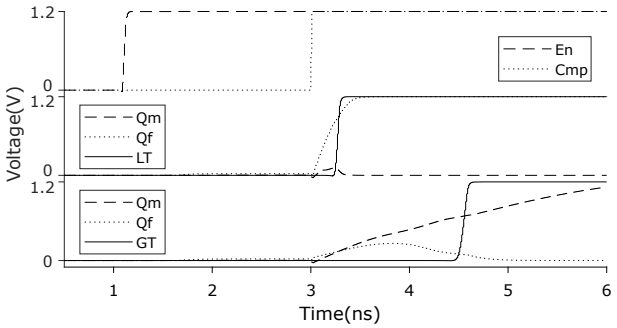

Fig. 13. The simulation result of the sample 264 with and without the offset calibration. The memristance is set as $280k\Omega$ which refers to the data 00. The DCCS is set as 00 which delivers less current than the selected memristance. Therefore, the expected result is the logic high at Q_m. However, the middle graph shows Q_m is low instead. After enabling $C_f = 10fF$, the signal is compensated and the correct result is shown in the bottom graph.

cially at $0°C$ and $85°C$. This is because the high memristance due to the low temperature causes insufficient voltage in the connected current mirror. In addition, the low memristance cannot dominate the impedance of the current mirror and results in the current saturation [50]. This needs the further investigation to find the proper solution.

The offset is also caused by the process variation. Fig. 12 shows the equivalent memristances that generate the same amount of current as the DCCS (data 00). It reveals the variation severely impacts the circuit because the memristance range is changed from $324.801k\Omega - 234.929k\Omega$ (Table VII) to $0 - 100k\Omega$. To solve this problem, the digitally programmable capacitor arrays can be attached to the latch's outputs [53], [54]. The capacitors: C_m and Cf have been connected to Q_m and Q_f through MN4 and MN5 in Fig. 7 to demonstrate this technique. The simulation based on the memristance of $280k\Omega$ and the Monte Carlo sample 264 ($-324.328k\Omega$ offset) without the compensation capacitances is illustrated in the middle graph of Fig. 13. The output of the circuit is LT because I_m is less than I_f due to the minus offset. Programming $C_f = 10fF$ can correct the result (GT is raised instead) as depicted in the bottom graph of Fig. 13.

B. Latency and Energy

The latency of the comparison process depends on the memristance. The result takes the longest time (worst) when the effective I_{Qm} and I_{Qf} are equal (maximum metastability). Therefore, the worst case latency is measured at the effective R_m in Table VII. On the contrary, the latency is shorter once the both currents are different. The best case latency is defined when the memristance is at the middle of two adjacent data boundaries. For example, the memristance of $280.550k\Omega$ gains the best case latency for data 00.

The latency is taken from the transition of En to the output of the MSR (GT/LT). Note that the lower bound of the data 11 is selected as $51.691k\Omega$ to keep the step size of $80k\Omega$ as explained in Section III-B. From Fig. 11(b), the latency of each case slightly decreases with the memristance as the higher current is generated. The best case latency ($< 2.36ns$) is approximately $550ps$ faster than that of the worst case ($<$

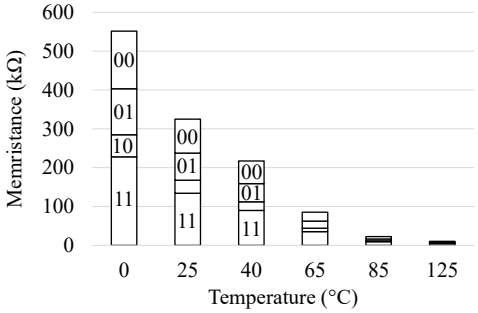

Fig. 14. The effect of temperature on each data boundary (2-bit). The data ranges are narrower when the temperature raises. They are likely at the same point once the temperature is at $85°C$. This is correlated to the proposed model simulation in Fig. 3

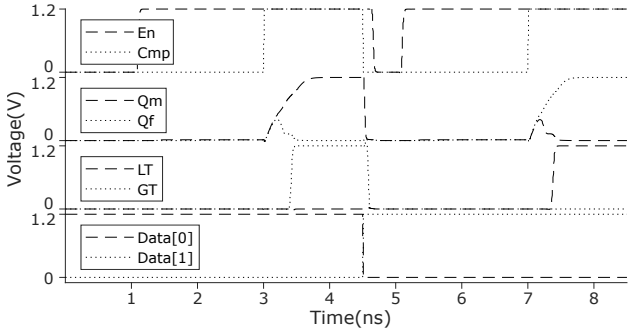

Fig. 15. Simulation result of decoding data 00 ($300k\Omega$) at $40°C$. The temperature changes R_m to $200.661k\Omega$ which is decoded as 01 instead.

$2.91ns$). The worst case latency limits maximum frequency. It depends on the input currents as depicted in Fig. 11(c). The input current at each data boundary increases when the memristance (data) is reduced. Regarding the worst case latency, the maximum frequency is calculated as $343MHz$.

The energy is measured from the rising edge of En signal to the rising edge of the MSR's output (GT/LT). The results in Table VII show the circuit's energy in worst case varies between $30.73fJ$ to $48.79fJ$. From the plot in Fig. 11(d), the energy rises in oppose to the memristance. This is because the lower memristance draws more current. The energy of the worst case scenario is higher than the best case one due to the longer latency caused by the metastability.

The maximum energy per comparison at each resolution is depicted in Fig. 11(e). The energy increases due to the higher number of active reference sources and the use of low memristance to support the higher resolution. In addition, the higher resolution requires more number of comparisons which results in the higher power consumption. Thus, the reading accuracy can be determined by the available energy. For example, if the decoder is powered from a stable power supply such as battery, the high precision (8-bit) can be selected. On the contrary, if the decoder is working using energy harvesting, the low energy precision (2-bit) can be selected instead.

C. Temperature Effect

The temperature affects the data ranges in Table VII as illustrated in Fig. 14. The ranges are wider at low temperature and become narrow when the temperature rises. For higher memristance values, the effect of temperature on linearity of the measured memristance is greater. Hence, smaller data values are more affected by temperature variation. This agrees with the modified model in (6) as the portion of R_{off}, which is highly temperature dependent and non-linear, is large at the small data (large memristance).

The temperature also causes a decoding error because the memristance and the comparator's references change at different rate. As an example in Fig. 15, initially, the memristance is specified as $300k\Omega$ at $25°C$. Therefore, this memristance stands in the range of 00 (Table VII). However, it drops to $200.661k\Omega$ which is decoded as 01 once the temperature rises to $40°C$. The compensation for the temperature effect is essential for devices working at different temperature conditions. This can be achieved by using our modified model shown in Section II in the controller.

V. Discussion

Because there is no resistance decoder reported previously, the performance of the resistance comparator which is the largest building block is considered and compared with the literature instead. The features of the related works are listed in Table VIII. For the proposed comparator, its performance shown in the table is selected from the worst results in Table VII. Furthermore, the power consumption is calculated from dividing the energy by the latency (time) of reading 11 which yields the highest value.

Although the comparator in [55] is faster than our work, it spends more power due to its static design. Therefore, it is not suitable for low-power applications. This is the common issue of any static comparators such as the works in [56] and [57].

The performance in [58] is prominent in the group of dynamic voltage comparator. It is also faster than our work while consumes less power. However, the bulk input design causes a leakage current to the predecessor stage. In addition, the design in [59] has approximately ten times longer delay while [60] requires a large area due to the use of resistors.

Static current comparators which indicate the direction of the input current are proposed in [56], [57]. The work in [56] employs a flipped voltage follower to lower the input impedance while [57] supports near-threshold operation with zero input offset. Nevertheless, their latency are very long, especially for [56]. Even though their power consumption figures are extremely low, they still exhibit power losses during the idle mode due to static operation. Furthermore, the power demands of their extra circuitry, such as current subtractor, are not included.

Another approach is translating the current to voltage using a differential transimpedance preamplifier (DTIA) which provides a better noise reduction according to the differential input design [38], [61]. The design in [61] is slightly faster than our work, but it has approximately 4 times higher power requirements. That design does not require extra circuitry, such as current subtractor, and can also operate at near-threshold voltage. However, it contains many large resistors which contributes area overhead and prone to the temperature

TABLE VIII
FEATURE COMPARISON.

Work	Voltage /Current	Static /Dynamic	Resistor included	Tech (nm)	Voltage (V)	Latency (ns)	Energy (fJ)	Power (uW)
[55]	V	S	N	65	1.0	0.20	N/A	95.00
[59]	V	D	N	180	1.2	14.97	147.00	N/A
[58]	V	D	N	180	1.2	1.84	N/A	18.60
[60]	V	D	Y	65	1.2	0.22	760.00	755.00
[56]	C	S	N	350	1.0	15.00	N/A	30.00
[57]	C	S	N	180	1.0	133.00	18.00	0.14
[61]	C	D	Y	180	0.5	2.20	N/A	79.00
[38]	C	D	N	180	1.8	0.95	N/A	697.00
This	C	D	N	65	1.2	2.91	46.23	21.13

and process variations. The design in [38] also implements DTIA. It is 3 times faster but needs 35 times higher power due to its static sub-circuits, such as subtractor and amplifiers. Note that the DTIAs in both works always consumes power because of their static operations.

The proposed decoder outperforms the literature which requires the extra circuitry to interface the memristor, control the power and support the power-accuracy scalability. Furthermore, its control interface supports both synchronous and asynchronous schemes. It is also resistor-free thus it can reduce the chip area and avoid any variations from the resistors.

VI. CONCLUSION

This paper proposes a realisation of a simple thermal model of TiO_2 memristor based on the recently proposed physical model and experimental data. The model evaluation shows the maximum error is 1.25% and the OFF memristance decreases quadratically and reaches ON memristance once the temperature closes to $85°C$. More simple, yet accurate model is open to be investigated in the future, as well as the parameters for the other materials.

We also propose a resistance decoder, which is based on a current-mode dynamic comparator to support power efficient and adaptive operation, for the memristance reading. Based on UMC 65nm low-leakage CMOS technology, the simulation results show the maximum offset of 1.70%, the worst case latency of $2.91ns$, the maximum frequency of $343MHz$ and the energy per comparison of $48.79fJ$ respectively. The proposed decoder supports a wide range of memristor applications 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 self-temperature compensation technique are subjects for future work.

REFERENCES

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

[2] 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.

[3] R. Shafik, A. Yakovlev, and S. Das, "Real-power computing," *IEEE Trans. Computers*, 2018.

[4] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E. Graves *et al.*, "Analogue signal and image processing with large memristor crossbars," *Nature Electronics*, vol. 1, no. 1, p. 52, 2018.

[5] 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.

[6] I. Vourkas and G. C. Sirakoulis, "Emerging memristor-based logic circuit design approaches: A review," *IEEE Circuits and Systems Magazine*, vol. 16, no. 3, pp. 15–30, 2016.

[7] A. Haj-Ali, R. Ben-Hur, N. Wald, R. Ronen, and S. Kvatinsky, "Not in name alone: A memristive memory processing unit for real in-memory processing," *IEEE Micro*, vol. 38, no. 5, pp. 13–21, 2018.

[8] 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 conf.* ACM, 2016, p. 19.

[9] Y. Bai, R. F. DeMara, J. Di, and M. Lin, "Clockless spintronic logic: A robust and ultra-low power computing paradigm," *IEEE Trans. Computers*, vol. 67, no. 5, pp. 631–645, 2018.

[10] S. Muthulakshmi, C. S. Dash, and S. Prabaharan, "Memristor augmented approximate adders and subtractors for image processing applications: An approach," *AEU-Int. J. Electronics and Communications*, vol. 91, pp. 91–102, 2018.

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

[12] R. Tetzlaff, *Memristors and memristive systems*. Springer, 2014.

[13] M. Chu, B. Kim, S. Park, H. Hwang, M. Jeon, B. H. Lee, and B.-G. Lee, "Neuromorphic hardware system for visual pattern recognition with memristor array and cmos neuron," *IEEE Trans. Industrial Electronics*, vol. 62, no. 4, pp. 2410–2419, 2015.

[14] A. K. Mukhopadhyay, I. Chakrabarti, A. Basu, and M. Sharad, "Power efficient spiking neural network classifier based on memristive crossbar network for spike sorting application," *arXiv preprint arXiv:1802.09047*, 2018.

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

[16] T. Bunnam, A. Soltan, D. Sokolov, and A. Yakovlev, "Pulse controlled memristor-based delay element," in *Int. Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, 2017.

[17] F. Puppo, M. A. Doucey, M. D. Ventra, G. D. Micheli, and S. Carrara, "Memristor-based devices for sensing," in *IEEE Int. Sym. Circuits and Systems (ISCAS)*, 2014, pp. 2257–2260.

[18] 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 J.*, vol. 15, no. 11, pp. 6208–6217, 2015.

[19] 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.

[20] 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.

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

[22] C. M. Vigorito, D. Ganesan, and A. G. Barto, "Adaptive control of duty cycling in energy-harvesting wireless sensor networks," in *Annual IEEE Communications Society Conf. Sensor, Mesh and Ad Hoc Communications and Networks*. IEEE, 2007, pp. 21–30.

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

[24] 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," *J. Applied Physics*, vol. 106, no. 7, p. 074508, 2009.

[25] C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, "Generalized memristive device spice model and its application in circuit design," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 8, pp. 1201–1214, 2013.

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

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

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

[29] 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 Trans. Multi-Scale Computing Systems*, 2017.

[30] 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.

[31] 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.

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

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

[34] 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 Trans. Electron Devices*, vol. 63, no. 5, pp. 1884–1892, 2016.

[35] I. Vourkas, *Memristor-based nanoelectronic computing circuits and architectures*. Springer, 2016.

[36] 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 Int. Sym. Circuits and Systems (ISCAS)*, 2015, pp. 185–188.

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

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

[39] O. Krestinskaya, A. Pappachen James, and L. O. Chua, "Neuromemristive circuits for edge computing: A review," *arXiv e-prints*, p. arXiv:1807.00962, 2018.

[40] 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 Trans. Nanotechnology*, vol. 15, no. 3, pp. 484–491, 2016.

[41] S. Kim, S.-J. Kim, K. M. Kim, S. R. Lee, M. Chang, E. Cho, Y.-B. Kim, C. J. Kim, U.-I. 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.

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

[43] 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.

[44] 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.

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

[46] B. Haaheim and T. G. Constandinou, "A sub-$1\mu w$, $16kHz$ current-mode sar-adc for single-neuron spike recording," in *IEEE Int. Sym. Circuits and Systems (ISCAS)*, 2012, pp. 2957–2960.

[47] 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.

[48] J. P. Carreira and J. E. Franca, "A two-step flash adc for digital cmos technology," in *Int. Conf. Advanced A-D and D-A Conversion Techniques and their Applications*, 1994, pp. 48–51.

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

[50] T. Bunnam, A. Soltan, D. Sokolov, and A. Yakovlev, "An excitation time model for general-purpose memristance tuning circuit," in *IEEE Int. Sym. Circuits and Systems (ISCAS)*, 2018, pp. 1–5.

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

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

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

[54] 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 Trans. Circuits and Systems I: Regular Papers*, vol. 59, no. 1, pp. 80–92, 2012.

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

[56] 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 J. Electrical and Computer Eng.*, vol. 39, no. 2, pp. 127–131, 2016.

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

[58] 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.

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

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

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

Thanasin Bunnam received his B.Eng. degree from Mahidol University, Thailand, in 2005 and M.Eng. degree from Chulalongkorn University, Thailand, in 2009. He was a lecturer at Rajamangala University of Technology Thanyaburi (RMUTT), Thailand, since 2013. He worked on ternary-logic asynchronous circuit design and IoT system implementation for agriculture and photovoltaic system. By support from the Royal Thai Government, he is currently a Ph.D. student at μSystem research group, School of Engineering, Newcastle University, U.K. His current research interests include memristor-based delay element, memristor-based multi-bit memory, memristor modelling and machine learning circuit design.

Ahmed Soltan received the B.Sc. and M.Sc. degrees from the University of Cairo, Cairo, Egypt, in 2004 and 2008, respectively. He received the Ph.D. degree in electronics and communication from Cairo University in 2014. He received the 2014-2016 best thesis award from Cairo for his Ph.D. thesis. He worked on circuit and system design and modelling in the fractional order domain during his Ph.D. research. He is currently a Research Associate and EDA/CAD Specialist with the School of Engineering, Newcastle University, Newcastle upon Tyne, U.K. He was a Teacher Assistant with the faculty of Engineering, Fayoum University, Fayoum, Egypt, for nine years and was an R&D Firmware Engineer for eight years. He also was an R&D Manager for an LED company at Qatar for one year and half. His current research interests include smart energy harvesting systems and power management for biomedical implantable devices and lab-on-chip systems. He is also interested in the thermal impact of the implantable devices on the human tissues, embedded system design for lab-on-chip system, the investigation of fractional circuits and systems, specifically in fractional order analog filters for signal processing, and fractional order modelling for biomedical applications. Dr. Soltan's research aims to establish a new healthcare monitoring system and diagnosis on the fly by development of autonomous devices. Dr. Soltan published more than 50 papers in a prestige journals.

Danil Sokolov (S'02-M'13) received the Ph.D. degree from Newcastle University, Newcastle upon Tyne, U.K., in 2006.

He is a Senior Research Associate with the School of Engineering, Newcastle University. From 2007 to 2010, he was a Consultant in an EDA start-up Elastix Corporation where he led research and development for ultra low power synthesis. He is the Lead Architect and a Developer of Workcrafta toolset for capture, simulation, synthesis, and verification of interpreted graph models. He has authored 16 journal articles, over 50 pair-reviewed conference papers, and a textbook chapter entitled System on Chip: Next Generation Electronics. His current research interests include modeling of self-timed heterogeneous systems, synthesis methods for energy-efficient circuits, and development of CAD tools and their integration into industry-level design flows.

Oleg Maevsky received his Ph.D. degree from St. Petersburg Electrical Engineering Institute, St. Petersburg, Russia, in 1987. In 1999 he was a Research Associate in Computing Science Department of University of Newcastle upon Tyne. From 2003 to 2007 he was a Sr. CAD-Engineer in Moscow group of Intel. From 2017 he is a director of Nanodevices LTD, Voronezh, Russia. He has authored about 45 journal articles and patents.

Alex Yakovlev (FIEEE, FIET) received his Ph.D. degree from St. Petersburg Electrical Engineering Institute, St. Petersburg, Russia, in 1982. He is a Professor of Computer Systems Design, who founded and leads the Microsystems Research Group, and co-founded the Asynchronous Systems Laboratory at Newcastle University. He was awarded an EPSRC Dream Fellowship in 2011-2013. He has published 8 edited and co-authored monographs and more than 300 papers in IEEE/ACM journals and conferences, in the area of concurrent and asynchronous circuits and systems, with several best paper awards and nominations. He has chaired organizational committees of major international conferences. He has been principal investigator on more than 30 research grants and supervised over 50 PhD students. He is a Fellow of Royal Academy of Engineering in the United Kingdom.