IEEE Journal on Exploratory Solid-State Computational Devices and Circuits

Received 22 August 2022; revised 1 October 2022 and 14 October 2022; accepted 19 October 2022. Date of publication 25 October 2022; date of current version 2 December 2022.

Digital Object Identifier 10.1109/JXCDC.2022.3217098

Memristive Devices for Time Domain Compute-in-Memory

FLORIAN FREYE [1], JIE LOU [1], CHRISTOPHER BENGEL [2], STEPHAN MENZEL [3] (Senior Member, IEEE), STEFAN WIEFELS [3], and TOBIAS GEMMEKE [1] (Senior Member, IEEE)

[1] Chair of Integrated Digital Systems and Circuit Design, RWTH Aachen University, 52074 Aachen, Germany
[2] Institut für Werkstoffe der Elektrotechnik 2, RWTH Aachen University, 52074 Aachen, Germany
[3] Forschungszentrum Jülich GmbH, Peter Gruenberg Institut (PGI-7), 52428 Jülich, Germany

CORRESPONDING AUTHOR: F. FREYE (freye@ids.rwth-aachen.de)

This work was supported in part by the Federal Ministry of Education and Research (BMBF) through the NEUROTEC Project under Project 16ME0398K and Project 16ME0399.

ABSTRACT Analog compute schemes and compute-in-memory (CIM) have emerged in an effort to reduce the increasing power hunger of convolutional neural networks (CNNs), which exceeds the constraints of edge devices. Memristive device types are a relatively new offering with interesting opportunities for unexplored circuit concepts. In this work, the use of memristive devices in cascaded time-domain CIM (TDCIM) is introduced with the primary goal of reducing the size of fully unrolled architectures. The different effects influencing the determinism in memristive devices are outlined together with reliability concerns. Architectures for binary as well as multibit multiply and accumulate (MAC) cells are presented and evaluated. As more involved circuits offer more accurate compute result, a tradeoff between design effort and accuracy comes into the picture. To further evaluate this tradeoff, the impact of variations on overall compute accuracy is discussed. The presented cells reach an energy/OP of 0.23 fJ at a size of 1.2 μm^2 for binary and 6.04 fJ at 3.2 μm^2 for 4 × 4 bit MAC operations.

INDEX TERMS Compute-in-memory (CIM), convolutional neural networks (CNNs), memristive devices, time-domain (TD) computing, time-domain CIM (TDCIM).

I. INTRODUCTION

SURPASSING the standing record in the ImageNet challenge by far, AlexNet started a continued surge in the use of convolutional neural networks (CNNs). A trend of ever-increasing network complexity is observed improving the accuracy while increasing the memory footprint and power consumption. To tackle both these challenges, new schemes for computation have emerged, which take inspiration from the human brain, i.e., the domain of neuromorphic computing. Thereby, one key principle is to compute-in-memory (CIM), an approach that co-locates data and computation to address the von Neumann bottleneck [1].

In this domain, analog computing is often considered to decrease power consumption further. The resilience of deep neural networks to a certain degree of imprecision is exploited with a moderate impact on the classification accuracy [2]. While current and charge domain computing have enjoyed high popularity for CIM [1], [3], [4], [5], they require expensive analog-to-digital conversion and lack in ability of voltage scaling. A different compute scheme is time-domain CIM (TDCIM). In time-domain (TD) computing, the values are encoded as discrete arrival times of signal edges. While signaling is sample discrete, the arrival time is fundamentally continuous. Similar to charge and current, time is inherently additive, allowing for efficient accumulation operations.

TD implementations vary, one example being the integration of currents on a capacitor and observing the reached voltage level [6], [7]. The total time is given using (1), with N as the number of multiply and accumulate (MAC) operations and I_{MAC} as the current component of a single MAC operation

$$t = \frac{C \cdot U}{\sum^{N} I_{\text{MAC}}}. \tag{1}$$

It becomes evident that for rising N, the time difference between MAC results diminishes, making time-to-digital conversion increasingly harder. Compensating this effect by increasing voltage, U, or output capacitance, C, comes with added cost. In this tradeoff, N is kept small, requiring more

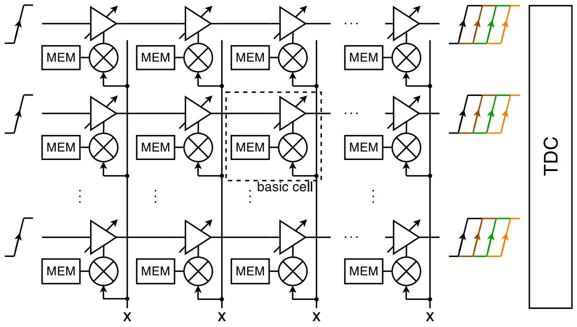


FIGURE 1. **Cascaded TDCIM array.**

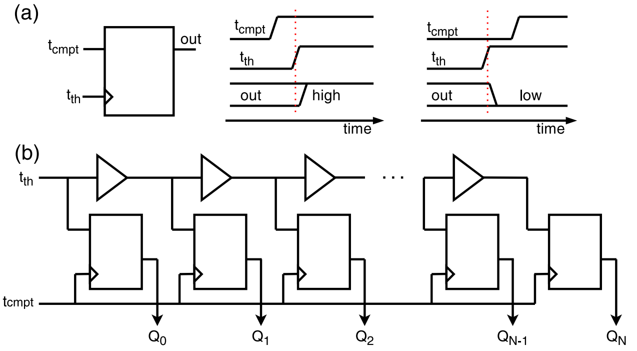


FIGURE 2. **TDC (a) binary and (b) multibit using threshold chain, t_{th}, and temperature-coded output, Q.**

time-to-digital conversions and accumulation of partial sums in the digital domain thereby increasing power consumption.

Cascaded TDCIM chains digitally adjustable delay elements to introduce a delay as a function of the multiplication result (see Fig. 1). The full-swing operation now offers margin for voltage scaling. Cascaded TDCIM is first introduced in [8] where registers are implemented by standard cell latches, allowing to unroll the complete kernel in a weight-stationary dataflow. However, the adopted standard cell memory suffers from large area. For a binary network implementation with an energy efficiency of 1.05 POPS/W, [9] devotes 22% of the total MAC area to memory. In [10], a cascaded TDCIM for ternary operation at 716 TOPS/W is shown using custom current-starved inverters combined with static random-access memory (SRAM). The used custom SRAM co-integrates the delay cell, leading to memory footprint of 50% of the total MAC area. For multibit operation, the increased memory size limits realizable chain length.

The designs in [11], [12], and [13] all use commercial SRAM separating memory and MAC operation. Due to the row sequential access of SRAMs, the chain size is limited by the number of SRAM columns. Therefore, these designs have short chain lengths of 128, 121, and 64, respectively. The two-terminal memristive switching devices can be used to replace SRAM to reduce the area. Among the different physical realizations, filamentary switching devices based on the valence change mechanism (VCM) are one of the most studied variants [14]. First memory macros have been introduced for embedded memory applications [15], [16]. Using such memory arrays, further application areas such as neuromorphic engineering have been demonstrated [17], [18]. Best area savings are achieved in monolithic 3-D structures placing the memristive devices in the back end of line (BEOL). Further area reduction potential results from their potentially high resistance leading to smaller caps in the delay elements. At the same time, their nonvolatile storage provides inherent leakage power savings.

On the downside, today's devices still feature high variability. As TDCIM is susceptible to process, temperature, and voltage (PVT) variations, prudent assessment of this is key. This work aims to assess these nonidealities and their impact on important design metrics to better understand the tradeoffs and limits that memristive devices entail in cascaded TDCIM. Section II gives an introduction into the general concept of memristive cascaded TDCIM. Section III introduces basic concerns of VCM reliability and mechanism of variability. Section IV presents a binary TDMAC cell based on VCM and discusses the implications of those nonidealities. In Section V, this concept is extended to the multibit case. Finally, we conclude in Section VI.

II. CASCADED TDCIM ARCHITECTURE

The operation of the typical convolution layer is shown in (2), where **x** is the input activation and **w** is the weight vector. f is the activation function and usually the binarize function for binary neural networks (BNNs) and the rectified linear unit (RELU) function for CNNs

$$\boldsymbol{Z} = f(\boldsymbol{w} \cdot \boldsymbol{x}). \tag{2}$$

In TD computing, cascaded variable delay elements can implement an accumulation. Each element realizes the delay to encode one multiplication result, thus realizing the MAC function. Unlike in the traditional digital circuits, the convolution result is therefore presented as an accumulated delay.

For the TDCIM architecture, the weights of one kernel are stored in the memory of one computing chain. For a kernel size of N with M computing chains in parallel, the total area, A_{tot}, is given by $M \cdot N \cdot A_{\text{cell}} + A_{\text{TDC}}$ with the cell area, A_{cell}, and the area for time-to-digital converter (TDC), A_{TDC}. After computing a convolution, the activation signals are changed, whereas the weights can remain in memory. The weights are only updated after the complete output feature map is computed, thus reducing the data movement from the main memory. The input activation can be shared by all the computing chains and further reduce the data movement.

The MAC computation is processed in the TD and will be converted into the digital domain by TDCs. For BNN, the TDC is reduced to a sampling of the output at a specific time-point using a standard flip-flop, as shown in Fig. 2(a) [9]. Based on the arrival time of the computing delay, t_{cmpt}, and the threshold delay, t_{th}, the binarized output, out, is generated. For multibit CNNs, sampling can also be performed using a threshold chain, producing a temperature-coded output [see Fig. 2(b)] [19]. To save area, sampling can alternatively be performed by an oscillator combined with a counter resulting in a tradeoff between area and sampling noise as variations get amplified by the number of oscillations. Using the same delay elements in TDC as for the compute chain ensures best attenuation of global chip variations. While it is not possible to subtract time, there are multiple ways to implement negative numbers in TD computing. By adding an offset to the delay, negative numbers can be represented by shorter delays than said offset, as done in [12] and [20]. In [8] and [21], a negative and a positive path are used with the numeric value being represented by the difference in delay.

By representing positive numbers with delays of a certain edge type and negative numbers with delays of the complementary edge type, duty cycle as an output with 50% marks the sign swap [22]. A basic technique to realize multibit numbers in TD lies in the bit-serial approach. Here, a multiplication can be split up into multiple multiplications with reduced word length down to binary. This way, negative values can be implemented by means of sign magnitude representation or one's/two's complement as done in [13].

A. MEMRISTIVE TDCIM

Memristive devices are used as the resistive component of the RC element in TD computing. The resulting difference in cell delay as a function of resistive states R_1 and R_2 is approximated according to (3). In a cascaded TD design, C is typically realized by the input capacitance of an inverter stage that follows the memristor:

$$\Delta t_{\mathrm{cell}} = 0.69(\tau_2 - \tau_1) = 0.69(R_2 - R_1) \cdot C. \tag{3}$$

As the numeric result is a function of programmed resistance values, any device-to-device or cycle-to-cycle variations in the memristive elements impact this delay value. As there are no write operations within the multiplication of a kernel with the input feature map, these two can be lumped to a single variation in resistance. An equivalent of the classical signal-to-noise ratio (SNR) can be found for TD computing in (μ_t/σ_t), with μ_t being the average delay step and σ_t being its standard deviation. Therefore, not only the variation in resistance but also the total resistance is important. Roughly a constant variation in current can be assumed for the low resistance state (LRS). For higher resistances within this state, this leads to an increase in the relative variation, as the nominal current, I_{nom}, goes down according to (4). This observation is also made in [23] for cycle-to-cycle variations and in [24] for chip-to-chip variations. For this reason, small delays can be realized with lower variations than higher delays

$$\frac{R}{R_{\mathrm{nom}}} = \frac{U_{\mathrm{nom}} \cdot I_{\mathrm{nom}}}{U_{\mathrm{nom}} \cdot (I_{\mathrm{nom}} + \Delta I)} = \frac{I_{\mathrm{nom}}}{I_{\mathrm{nom}} + \Delta I}. \tag{4}$$

In the binary case, the high resistance state (HRS) is programmed to realize long delays. As long delays introduce higher total error for the same relative error, HRS variability bounds binary accuracy. On the other hand, resistances for multibit lie within the LRS to allow for multiple steps of resistance. Thus, multibit accuracy is bound by LRS variability.

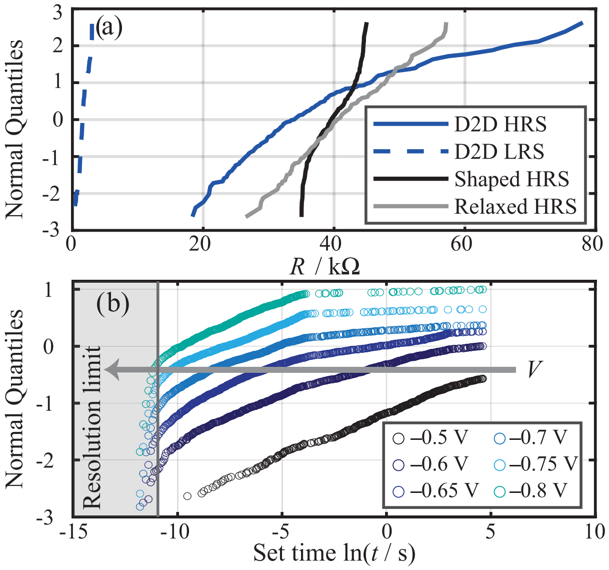


FIGURE 3. Experimental characteristics of ZrO_2 VCM cells. (a) Resistance distributions. (b) SET kinetics redrawn from [39].

Besides the effects influencing variability, reliability aspects have to be considered. These include read disturb, writability, and forming and will be addressed in the next section.

III. VARIATIONS AND RELIABILITY CONCERNS OF VCM

The VCM cells consist in their simplest form of a metal oxide (e.g., TiO_2, ZrO_2, HfO_2, Ta_2O_5, or $SrTiO_3$) sandwiched between two different metal electrodes [25]. The switching mechanism in VCM cells is based on the movement of oxygen vacancy defects within the oxide region [26], [27]. Prior to the repetitive switching, a so-called electroforming process is required, during which the oxygen vacancies are introduced into the system via an oxygen exchange process at the metal/oxide interfaces [28]. While it only happens once, the variability of this process may influence the later switching performance and can already lead to device-to-device variability [29]. Due to the switching variability also the programmed states show some variability as illustrated in Fig. 3(a). Whereas the programmed high resistive state shows a log-normal distribution, the programmed low resistive state shows a normal distribution of the resistance states. This behavior is typical for filamentary VCM cells [30], [31], [32]. It was observed that the states relax after programming, which leads to a widening of the distribution [33], [34], [35]. Moreover, it was shown that reprogramming the tail bits (i.e., shaping the distribution) by applying an additional programming pulse has no effect as the stable distribution is restored after some time [30], [34], [35]. Another variability aspect of concern is the state retention, i.e., the change in the resistance over time. It has, however, been shown that the states are stable over few hundred hours at temperatures above 150 °C [36], [37], [38].

Read disturb describes the directed change in a resistance distribution under a repeated application of read pulses. It therefore not only represents a variability concern but one for reliability as well. So far, read disturb properties have been mostly investigated for binary devices and under the assumption of constant voltage during each read pulse. Under these assumptions and for the use case of an embedded memory, recent results have shown that read disturb is not a critical issue in advanced integrated technologies [36]. In [39], we have demonstrated read disturb stability for binary devices under constant read voltage pulses for up to 5×10^{10} VMM operations suggesting that far larger numbers of read operations are possible than the conventionally investigated for memory applications under the right conditions. For this stability, the SET voltage has to be kept below −0.3 V and the RESET voltage has to be kept below 0.5 V. While short read pulses can be expected for cascaded TDCIM, further investigations must be made on read disturb, as shape and duration of the read pulse heavily depend on circuit implementation.

Besides read disturb, other reliability aspects have to be investigated for the use of memristors. Due to the ionic nature of the switching mechanism, the switching process is inherently stochastic. Owing to the small dimensions of the filament, Joule heating occurs in this device accelerating the switching process further. On one hand, Joule heating is the key enabler solving the voltage–time dilemma, i.e., high stability at small read voltages while enabling fast switching times at high write voltages [40]. On the other hand, Joule heating introduces a positive feedback during switching [41], leading to a strong state dependence of the switching time. It has been shown that the switching time can vary over orders of magnitudes for a given voltage from cycle-to-cycle and device-to-device [42], [43]. In consequence, one can expect to have slow switching devices and fast switching devices in an array, as illustrated in Fig. 3(b). Tuning for good writability therefore also increases susceptibility to read disturb. The spread of switching times has further implications on the writing process. In principle, the programming pulsewidth could be chosen long enough for successful one-shot programming. However, most devices will then switch early and a high amount of energy is dissipated. Moreover, the fast switching devices may be overprogrammed leading to failed rewrite attempts. Thus, in most cases, a so-called write-verify process is used to program desired resistance states [33], [44]. Le et al. [33] demonstrate such programming of eight different resistance states on a large array. Resistance relaxation or retention describes changes in the programmed resistance distributions and therefore also introduces reliability concerns. In contrast to read disturb, it is neither directed to a certain resistance state nor directly associated with repeated reading as exhibited, e.g., by the relaxed HRS distribution in Fig. 3(a). In [33], the retention properties of 3-bit VCM cells were investigated in which the bit error rate (BER) was increased from 0% to 0.6% after the experiment. As the relaxation is stronger at higher, resistances the resistances were all kept below 35 kΩ [33].

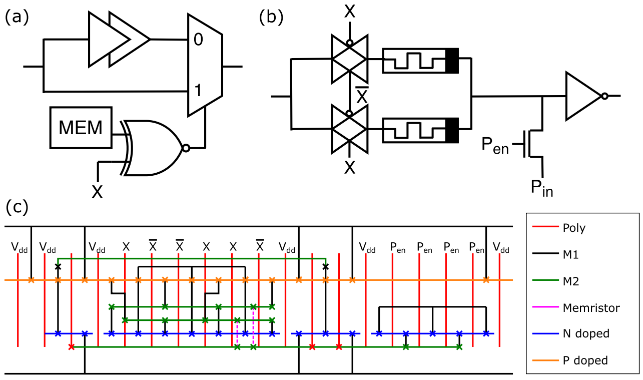


FIGURE 4. Binary MAC cell. (a) CMOS only. (b) Memristive-based. (c) Stick diagram of the memristive-based cell for area estimation.

The maximum required voltages to operate the devices are important as they have to be supported by the transistors. As the initial forming step requires the highest voltages applied for the longest time, it is the most critical one in that regard. Different proposals have been made to tackle this problem such as implanting the oxide of the VCM devices [45] or adapted pulsing schemes [46]. In [47], the use of an additional deep n-well allowed for keeping all the applied voltages within the limitations of the core devices (1 V), while still allowing for high enough voltages for the forming process (1.65 V). In addition, the gap between forming voltage and technology node has been shrinking from around 2 V at 130 nm to about 1 V at 14 nm [47]. Compared with bulk devices, fdSOI offers elevated drain-source breakdown voltage (BVDS) with [48] reporting more than 2 V for soft breakdowns in a 22-nm process. Therefore, elevated voltages for the forming step can be tolerated, as this step is done only once, and time-dependent degradation therefore is negligible.

To investigate the variability and reliability of filamentary VCM cells as shown in Fig. 3(a), the respective devices with a 30-nm Pt/5-nm ZrO_2/20-nm Ta/30-nm Pt stack were fabricated into a 7 × 7 μm crossbar architecture. The Pt bottom electrode is connected to ground and all voltages are applied to the Ta/Pt top electrode. However, for the sake of comparability, all the voltages in this article are given with respect to 0 V at the Ta/Pt top electrode. Further details on the device fabrication and the measurement setup can be found in [39].

IV. PROPOSED MAC CELL FOR BNNs

Fig. 4(a) shows implementations for binary TDCIM MAC cells in classical CMOS (a) compared with a memristor-based solution (b) and the corresponding stick diagram (c). Both designs allow for computation on rising and falling edges, increasing the energy efficiency. For BNNs, weights are typically defined as −1 and 1 as presented in [49]. The values −1 and 1 can be mapped to 0 and 1, thus translating an XNOR operation to a multiplication. In the CMOS case, the XNOR gate therefore implements multiplication. The result of the multiplication is then connected to the variable delay element consisting of a multiplexer and a delay cell. In the memristive implementation, the memory, multiplication, and variable delay are not that clearly separable. A construct of two complementary controlled transmission gates with two complementary programmed memristors acts as a resistive XNOR gate. For $W = 1$, the lower memristor is in HRS and the upper memristor is in LRS, respectively. Thus, $X = 1$ activates R_{HRS} in the delay path and $X = 0$ activates R_{LRS}. For $W = 0$, the memristors are swapped, leading to an inverted operation with respect to X. The resistance of this gate holds the weight, but also realizes the delay, as it implements an RC-element when considering the gate capacitance of the output inverter. The NMOS before the output inverter is used for programming.

For the memristive implementation, an inverting design is shown whereas the CMOS example is noninverting. This only leads to a swap of trigger direction in TDC in case of an odd number of delay elements for the inverting case.

A. VARIABILITY ANALYSIS

The delay of individual MAC cells is sensitive to noise and PVT variations. Here, process variations can be separated into global or chip-to-chip variations as well as local variations, which are present on a single chip. For TDCs build from the same delay cells used for computations, global variations have the same impact on all the delay paths. Thus, accuracy of the computation is only susceptible to local variations [50] considering a matched TDC circuit [9].

The compute chain SNR is directly related to the SNR of a single cell. As the delay of the individual stages accumulates over the course of the compute chain, σ_{chain} can be obtained by (5), with N being the length of the compute chain and μ_i and σ_i the mean and standard deviation of the ith cell, respectively. Thus, the SNR of the computation is given by (6)

$$\sigma_{\mathrm{chain}} = \sqrt{\sum_i^N \sigma_i^2} \tag{5}$$

$$\mathrm{SNR}_{\mathrm{chain}} = \frac{\mu_{\mathrm{chain}}}{\sigma_{\mathrm{chain}}} = \frac{\sum_i^N \mu_i}{\sqrt{\sum_i^N \sigma_i^2}}. \tag{6}$$

Due to σ_{chain} growing in a square root relationship to the chain length, longer chains generally provide a better SNR. For the binary case, the equation can be simplified to the following equation by assuming the MAC cells delay step size as Δt:

$$\mathrm{SNR}_{\mathrm{chain,bin}} = \frac{\sum_i^N \Delta t}{\sqrt{\sum_i^N \sigma_{\Delta t}^2}} = \mathrm{SNR}_{\mathrm{cell}} \cdot \sqrt{N}. \tag{7}$$

In [51], we use this relationship to obtain the mean square error (mse). The central limit theorem allows to assume the Gaussian variations for the compute chain, leading to the following equation:

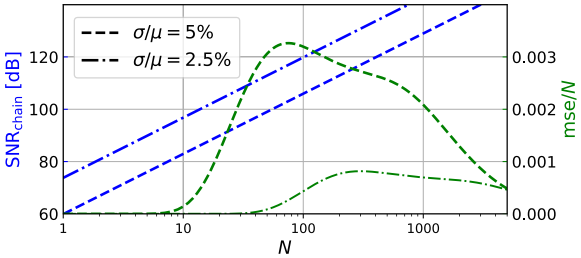


FIGURE 5. SNR and relative mse over compute chain length.

$$\mathrm{mse} = \sum_i^N i^2 \left[\mathrm{erf}\left((i+0.5)\frac{\mathrm{SNR}_{\mathrm{cell}}}{\sqrt{2N}} \right) - \mathrm{erf}\left((i-0.5)\frac{\mathrm{SNR}_{\mathrm{cell}}}{\sqrt{2N}} \right) \right]. \tag{8}$$

Plotting (mse/N) in Fig. 5 reveals a regime, where mse is zero. Here, the compute chain length is sufficiently short that the error is smaller than the threshold for the next value. The threshold for this regime is given by the following equation (9) [51]:

$$\frac{1}{6}\frac{\sqrt{N}}{N+1} = \frac{1}{\mathrm{SNR}_{\mathrm{cell}}}. \tag{9}$$

In this regime, the TD computation can be assumed to be purely deterministic. Usually, operation does not take place in this regime as it only allows for short compute chains or large delay increments, and therefore low energy efficiency.

Due to the HRS providing a lower current, classical crossbar vector multipliers are specially sensible to the variations within the LRS state. For binary TD computing, this relationship is reversed, as the HRS has a higher time constant. To model the influence of these variations compared with process variations seen in classical transistors, the PDK of a commercial 22-nm fdSOI technology was used to obtain transistor device models including back-annotated variability. Memristors are modeled with a resistance of configurable process variability.

In contrast to other compute schemes, cascaded TD computing offers good voltage scaling capabilities. By scaling voltage, efficiency is traded off against lower SNR due to higher impact of transistor threshold voltage variations. In Fig. 6, we provide cell level SNR for different voltage levels and different combinations of relative memristor variance as well as HRS resistances. For low memristor variations, transistor variations significantly contribute to cell variations, leading to more than 2% cell variance for $\sigma_{R_{\mathrm{HRS}}} = 1\%$. For scaled voltages, this effect is amplified, leading to cell variations of more than 4%. For a higher noise assumption, the memristor variance dominates and this effect diminishes. Here, we see that higher R_{HRS} delivers better SNR. This can be explained with an increase in μ, while memristor variation remains constant over voltage. The red line indicates the HRS case from the tuned case in Fig. 3(b). The assumed LRS

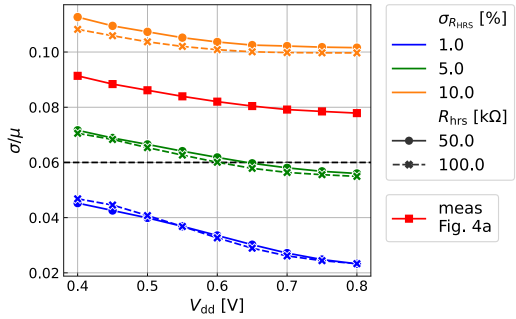


FIGURE 6. **Binary MAC cell delay variability for changing memristor variability, $\sigma_{R_{HRS}}$, and HRS resistance, R_{HRS}.**

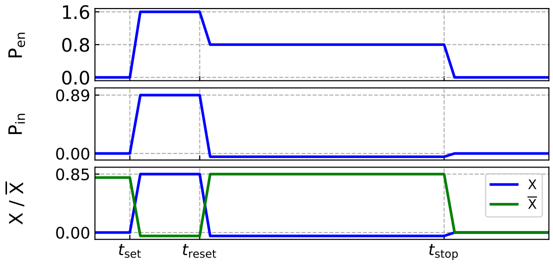
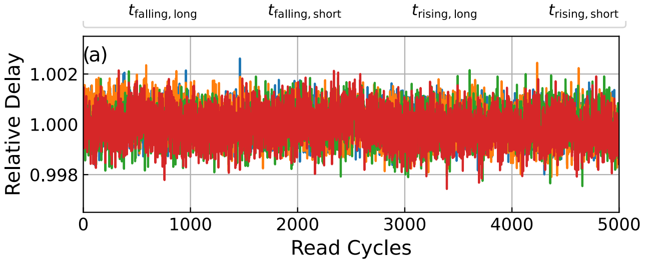


FIGURE 7. **Write scheme visualization.**



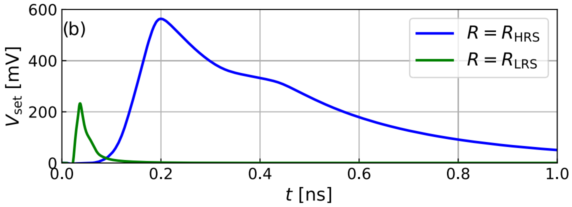


FIGURE 8. **(a) Delay variation of a single cell over 5000 read cycles. (b) Differential voltage over the memristor.**

resistance is 1.5 kΩ, in correspondence to the measurements above. Lower mean resistance leads to a similar SNR as the example with higher memristor variations but higher mean.

In [51], we show that (σ/μ) below 6% has negligible impact on network accuracy for the MNIST dataset. While this threshold is surpassed for the tuned HRS case, for $\sigma_{R_{HRS}} = 5\%$, the requirement is fulfilled even for voltage scaling to 0.6 V.

B. WRITE SCHEME AND READ DISTURB

Besides variations, writability and forming of the memristors are concerns when combined with core devices, as high voltage is typically required. For forming, elevated voltages are assumed to be tolerable, as this step is performed only once. Tuning for easier writability trades off with susceptibility to read disturb. Therefore, these concerns are closely related.

Fig. 7 shows the write scheme for the proposed cell. The inverting property is used, so that setting the chain-wide programming enable P_{en}, pulls all the inverter stage inputs to P_{in} resulting in a differential voltage over the resistive XNOR gate. Altering X allows a selection of the memristors to be written. To reduce the voltage drop over the transistor, fdSOI back-gate biasing is used, and all the transistors involved in programming are super low threshold devices. In addition, the programming signals and the control signals for X can be chosen higher than V_{dd}, as the voltage drop over the transistor ensures safe operating margins for all the devices. This way, the differential programming voltage can be increased. After the chain-wide write procedure, a write-verify process can be implemented for the HRS by controlling the X input cell-by-cell.

To test for writability, the Verilog-A model presented in [30] (JART VCM v1b) was used with a reduction of the maximum oxygen vacancy density of $N_{plug} = N_{disk,max} = 4 \times 10^{26}$ m^3. In this configuration, the HRS resistance, R_{HRS}, was determined as 150 kΩ for this configuration, therefore generating an even longer read voltage pulse than previously. To find corners for fast and slow switching devices, the model parameters l_{disk}, r_{disk}, N_{min}, and N_{max} were altered, which represent the length of the disk region, the radius of the filament, and the minimum and maximum oxygen vacancy densities, respectively. To analyze writability, all the parameters were therefore altered by ±10%. Writability was confirmed in all the corners.

To ensure read stability, 5000 pulses equivalent to 10 000 computations were applied to the TD MAC cell. Here, the fast switching corner was set up in an effort to create a realistic edge case. The relative cell delay shows no systematic drift and only shows small noise indicating no read disturb issue even for a supply voltage of 0.8 V [see Fig. 8(a)]. This can be accounted to really short differential voltage over the memristor [see Fig. 8(b)] which represents another benefit of cascaded TDCIM.

C. CELL COMPARISON

Table 1 shows a comparison of binary MAC cells for cascaded TDCIM. All the cells can implement unrolled kernels to allow for minimal memory movement. While [10] provides a small footprint when scaled, its delay element only consists of a single current-starved inverter, leading to low SNR. For the sake of comparability, a version of the design was considered, which queues multiple delay stages, scaling the SNR by $(N_{scale})^{1/2}$. For $N_{scale} = 4$, a similar (σ_t/μ_t) to this work is reached.

The design presented in [9] is optimized for SNR at the cost of area. Therefore, the area-sensitive DLY40 version

TABLE 1. Comparison of cascaded and unrolled TDCIM cells.

	[10]	[10]*	[9]	This work
Type	ternary	ternary	binary	binary
Node	40 nm	40 nm	22 nm	22 nm
Area†	1.5 µm^2	3.75 µm^2	3.08 µm^2	1.2 µm^2‡
$\frac{\sigma}{\mu}$	0.16	0.08	0.041@0.8 V 0.178@0.4 V	0.078@0.8 V 0.091@0.4 V
Δt	437 ps (258 ps†)	1.75 ns (1 ns†)	71 ps	86.5 ps
$\frac{\text{Energy}}{OP}$	1.4 fJ (0.55 fJ†)	5.6 fJ (2.2 fJ†)	1.22 fJ@0.8 V 0.26 fJ@0.4 V	1.26 fJ@0.8 V 0.23 fJ@0.4 V

*scaled to 6 delay elements; †scaled to 22 nm @0.8 V; ‡estimation based on stick diagram in Fig. 4c

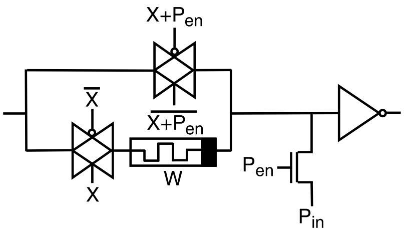


FIGURE 9. Multibit MAC cell schematic.

evaluated in simulation was used for comparison. Area estimation is performed using the stick diagram from Fig. 4(c) by assuming eight-track cell height corresponding to the used technology and a width of 18 contact poly pitches of 100 nm [52]. Accounting for technology scaling, the memristive implementation provides a 2× smaller footprint to [9] and a 3× smaller footprint to [10] adapted for comparable SNR. Comparing SNR at scaled voltages reveals another major advantage of implementing cascaded TDCIM using memristors. When V_{dd} approaches the transistor threshold voltage, the variations in threshold voltage get amplified, decreasing SNR. Contrary to the transistor, the memristor read behavior remains linear for lower voltages, leading to less deterioration of SNR. In this comparison, the setup corresponding to the red curve from Fig. 6 is assumed.

Given the same die area, the throughput of the designs is mainly dominated by two parameters: cell area and delay step size, Δt. Minimizing the former allows for greater parallelism and therefore better throughput. Using wave pipelining as shown in [9] computations can be overlapped, speeding up the time per computation in a chain from $t_{\min} + \Delta t$ to Δt. By considering $\frac{1}{\text{Area}\Delta t}$ as a figure of merit for the throughput, the presented design also proves advantageous.

V. MEMRISTIVE MULTIBIT TDCIM

A. MAC CELL IMPLEMENTATION

Multibit TDCIM is mostly implemented in a bit-serial fashion. Here, MAC cells implement a 1-bit by X-bit multiplication and intermediate results are combined by a shift and add operation. Slight changes to the MAC cell presented in Section IV realize such an implementation (see Fig. 9). For $X = 0$, the memristor is bridged by the upper TX gate and blocked by the lower one. For $X = 1$, the TX gates change roll and the memristor is connected in series with the output inverter. Here, delay depends on the memristor state. By combining two compute chains, a sign-magnitude implementation is realized. To realize positive numbers, the memristors are programmed with $(|W| + 1) \cdot R_{\text{step}}$ in the positive chain and R_{step} in the negative chain. Negative numbers are implemented by switching this assignment for a time difference of $W \cdot \Delta t_{\text{step}}$ between both the chains. For TDC + RELU, the negative path can be used as the threshold chain input in Fig. 2.

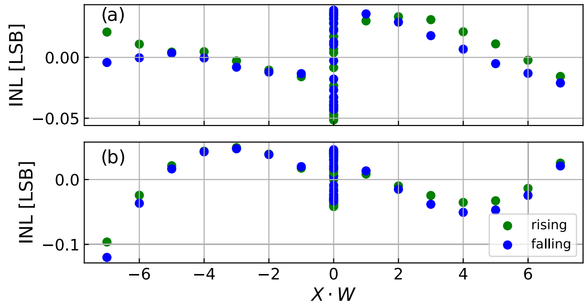


FIGURE 10. Multibit INL for (a) $\sigma_{\text{LRS}} = 1\%$ and (b) $\sigma_{\text{LRS}} = 2\%$.

The write scheme for the multibit version can be directly copied from the binary case due to the similarity in design. To prevent stress on the upper TX gate, P_{in} can, however, not be globally put above 0.88 V and may only be increased for cells which are supposed to be written in set direction. Findings on reliability in Section IV can be applied to the multibit MAC cell as, besides of the case $X = 0$, the same devices are involved in writing and reading the circuit.

Due to the additional logic controlling the upper transmission gate, the area increases by the size of a NOR gate and an inverter, equivalent to 0.4 µm^2. Together with the negative chain, the area of the multibit design is $2 \times (1.2 + 0.4\ \mu\text{m}^2) = 1.66 \cdot A_{\text{cell}}$. Energy consumption increases linearly with the word length of the activation, resulting in 10.8 fJ for the complete 4 × 4 MAC operation. Within this section, R_{LRS} is assumed as 15 kΩ, resulting in a unit delay step of 27.5 ps. Throughput is limited by the maximum cell delay, $t_{\max}$, which is 257 ps for $W = 7$. At 0.7-V supply voltage, the energy/Op reduces to 6.04 fJ and $t_{\max}$ increases to 265 ps.

B. NONLINEARITY AND VARIATIONS

In contrast to the binary case, linearity is a concern for the multibit case, adding onto variations and noise. Besides nonlinearities in the transmission gates and the Miller effect, another effect influences linearity proportional to memristor variance. As the delay is reciprocal to the conductance of the memristor, negative deviations in conductance influence the delay to a higher degree than deviations of same amplitude in the other direction, hence shifting the mean value. Due to the relationship in (4), higher delay values have higher variance and therefore are influenced to higher degree by this effect. Comparing integral nonlinearity (INL) of $\sigma_{\text{LRS}} = 1\%$ and $\sigma_{\text{LRS}} = 2\%$, this becomes obvious, as for $\sigma_{\text{LRS}} = 2\%$ INL increases (see Fig. 10).

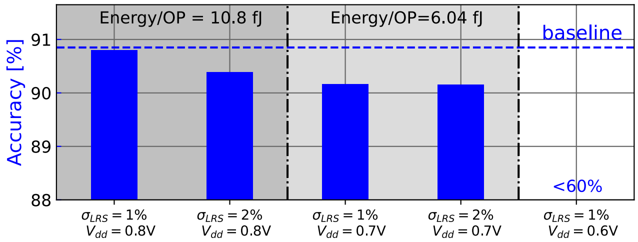


FIGURE 11. **Network accuracy evaluation.**

Unwanted deviations in the delays for $X = 0$ can be noted. To ensure writability, the memristor is only isolated from the input direction, leading to current flowing into the parasitic capacitance of the closed transmission gate. Thus, a spread in delay values is observed for $X = 0$.

C. NETWORK ACCURACY

Different combinations of X and W will have different standard deviations, leading to a complex relationship between error and chain length. The error of a single computation, e, can be estimated using (10), and the resulting error after the shift&add operation can then be modeled with (11)

$$\mu_{\text{chain}} = \sum_i^N \text{inl}(x_i, w_i), \quad \sigma_{\text{chain}} = \sqrt{\sum_i^N \sigma^2_{\text{cell}}(x_i, w_i)} \quad (10)$$

$$e = \sum_i^B 2^i \cdot \mathcal{N}(\mu_{\text{chain}}, \sigma_{\text{chain}}). \quad (11)$$

To estimate the resulting effect on network accuracy, the error model of (11) was applied to resnet20 on the CIFAR10 dataset. For the quantized networks, the first layer and the last layer were kept at 8 bit without added noise and the TDC is assumed to be sufficiently accurate. The results for INL and σ_{cell} were obtained by averaging falling and rising edge results. The achieved accuracies after training are shown in Fig. 11. For $V_{\text{dd}} = 0.8$ V, the memristor variation dominates, leading to a drop in accuracy from $\sigma_{\text{LRS}} = 1\%$ to $\sigma_{\text{LRS}} = 2\%$. For scaled voltages, the transistor variance increases and acts as the new bottleneck, leaving only a small difference between both the variation levels. Voltage scaling below 0.7 V was not attainable as the overall noise increases too much. A method to allow further voltage scaling could lie in increasing R_{LRS}, hence sacrificing throughput for improved SNR.

VI. CONCLUSION

In this work, the use of memristive devices for TDCIM is evaluated. Thereby, benefits of the cascaded approach for TD computing based on these devices offer promising alternatives to classical memory especially considering area reduction. Variability and reliability aspects of memristive devices were discussed in the context of TDCIM applications. An implementation for a binary TDCIM MAC cell is presented, and rigorous analysis on the impact of variations and reliability was performed. While the reached SNR is still not fully competitive to pure CMOS implementation at regular supply voltages, all other design goals could be met or surpassed for the memristive implementation. For reduced supply voltages, the memristive implementation outperforms even in terms of SNR. We expect that improvements in manufacturing quality soon will close this gap, enabling highly competitive memristive TD implementations. The limits on tolerated variations to achieve this goal were derived for the binary case.

In addition to the binary case, the TDCIM MAC cell was altered to support multibit operation. The proposed cell introduces minimal overhead in size using the shift&add operations and offers comparable reliability. An error model is presented to obtain network accuracy estimates for designs using nonlinearities and variations and is used to evaluate network performance. For $\sigma_{\text{LRS}} = 1\%$, these nonidealities could be mitigated in training, almost reaching classification accuracy to the purely quantized network. While the presented design shows less headroom for voltage scaling, it is well-suited for increasing throughput and reducing area.

REFERENCES

[1] V. Sharma, J. E. Kim, Y.-J. Jo, Y. Chen, and T. T.-H. Kim, "AND8T SRAM macro with improved linearity for multi-bit in-memory computing," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5.

[2] C.-Y. Chen, J. Choi, K. Gopalakrishnan, V. Srinivasan, and S. Venkataramani, "Exploiting approximate computing for deep learning acceleration," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2018, pp. 821–826.

[3] Z. Chen, Q. Jin, J. Wang, Y. Wang, and K. Yang, "MC2-RAM: An In-8T-SRAM computing macro featuring multi-bit charge-domain computing and ADC-reduction weight encoding," in *Proc. IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED)*, Jul. 2021, pp. 1–6.

[4] X. Si et al., "A 28 nm 64Kb 6T SRAM computing-in-memory macro with 8b MAC operation for AI edge chips," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 246–248.

[5] M. E. Sinangil et al., "A 7-nm compute-in-memory SRAM macro supporting multi-bit input, weight and output and achieving 351 TOPS/W and 372.4 GOPS," *IEEE J. Solid-State Circuits*, vol. 56, no. 1, pp. 188–198, Jan. 2021.

[6] M. Bavandpour, S. Sahay, M. R. Mahmoodi, and D. Strukov, "Efficient mixed-signal neurocomputing via successive integration and rescaling," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 3, pp. 823–827, Mar. 2020.

[7] S. Sahay, M. Bavandpour, M. R. Mahmoodi, and D. Strukov, "Energy-efficient moderate precision time-domain mixed-signal vector-by-matrix multiplier exploiting 1T-1R arrays," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, vol. 6, pp. 18–26, 2020.

[8] D. Miyashita, S. Kousai, T. Suzuki, and J. Deguchi, "A neuromorphic chip optimized for deep learning and CMOS technology with time-domain analog and digital mixed-signal processing," *IEEE J. Solid-State Circuits*, vol. 52, no. 10, pp. 2679–2689, Oct. 2017.

[9] J. Lou et al., "All-digital time-domain compute-in-memory engine for binary neural networks with 1.05 POPS/W energy efficiency," in *Proc. Eur. Solid-State Circuits Conf. (ESSCIRC)*, 2022, pp. 149–152.

[10] J. Song et al., "TD-SRAM: Time-domain-based in-memory computing macro for binary neural networks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 8, pp. 3377–3387, Aug. 2021.

[11] J. Yang et al., "TIMAQ: A time-domain computing-in-memory-based processor using predictable decomposed convolution for arbitrary quantized DNNs," *IEEE J. Solid-State Circuits*, vol. 56, no. 10, pp. 3021–3038, Oct. 2021.

[12] L. R. Everson, M. Liu, N. Pande, and C. H. Kim, "An energy-efficient one-shot time-based neural network accelerator employing dynamic threshold error correction in 65 nm," *IEEE J. Solid-State Circuits*, vol. 54, no. 10, pp. 2777–2785, Oct. 2019.

[13] P.-C. Wu et al., "A 28 nm 1 Mb time-domain computing-in-memory 6T-SRAM macro with a 6.6 ns latency, 1241GOPS and 37.01TOPS/W for 8b-MAC operations for edge-AI devices," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2022, pp. 1–3.

[14] A. Chen, "A review of emerging non-volatile memory (NVM) technologies and applications," *Solid-State Electron.*, vol. 125, pp. 25–38, Nov. 2016.

[15] A. Hayakawa et al., "Resolving endurance and program time trade-off of 40 nm TaO_x-based ReRAM by co-optimizing verify cycles, reset voltage and ECC strength," in *Proc. IEEE Int. Memory Workshop (IMW)*, May 2017, pp. 1–4.

[16] Y.-C. Chiu et al., "A 40 nm 2 Mb ReRAM macro with 85% reduction in FORMING time and 99% reduction in page-write time using auto-FORMING and auto-write schemes," in *Proc. IEEE Symp. VLSI Technol.*, Jun. 2019, pp. T232–T233.

[17] M. Giordano et al., "CHIMERA: A 0.92 TOPS, 2.2 TOPS/W edge AI accelerator with 2 MByte on-chip foundry resistive RAM for efficient training and inference," in *Proc. Symp. VLSI Circuits*, Jun. 2021, pp. 1–2.

[18] H. Li et al., "SAPIENS: A 64-kb RRAM-based non-volatile associative memory for one-shot learning and inference at the edge," *IEEE Trans. Electron Devices*, vol. 68, no. 12, pp. 6637–6643, Dec. 2021.

[19] K. Kim, W. Yu, and S. Cho, "A 9 bit, 1.12 ps resolution 2.5 b/stage pipelined time-to-digital converter in 65 nm CMOS using time-register," *IEEE J. Solid-State Circuits*, vol. 49, no. 4, pp. 1007–1016, Apr. 2014.

[20] D. Miyashita et al., "An LDPC decoder with time-domain analog and digital mixed-signal processing," *IEEE J. Solid-State Circuits*, vol. 49, no. 1, pp. 73–83, Jan. 2014.

[21] S. Gopal et al., "A spatial multi-bit sub-1-V time-domain matrix multiplier interface for approximate computing in 65-nm CMOS," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 8, no. 3, pp. 506–518, Sep. 2018.

[22] J. Yang et al., "Sandwich-RAM: An energy-efficient in-memory BWN architecture with pulse-width modulation: Digest of technical papers," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 394–396.

[23] C. Giovinazzo et al., "Analog control of retainable resistance multistates in HfO_2 resistive-switching random access memories (ReRAMs)," *ACS Appl. Electron. Mater.*, vol. 1, no. 6, pp. 900–909, 2019.

[24] R. Yasuhara et al., "Reliability issues in analog ReRAM based neural-network processor," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Mar. 2019, pp. 1–5.

[25] J. J. Yang and R. S. Williams, "Memristive devices in computing system: Promises and challenges," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 9, no. 2, pp. 1–20, May 2013.

[26] R. Waser, R. Dittmann, G. Staikov, and K. Szot, "Redox-based resistive switching memories—Nanoionic mechanisms, prospects, and challenges," *Adv. Mater.*, vol. 21, nos. 25–26, pp. 2632–2663, Jul. 2009.

[27] J. J. Yang, M. D. Pickett, X. Li, D. A. A. Ohlberg, D. R. Stewart, and R. S. Williams, "Memristive switching mechanism for metal/oxide/metal nanodevices," *Nature Nanotechnol.*, vol. 3, no. 7, pp. 429–433, Jul. 2008.

[28] J. J. Yang et al., "The mechanism of electroforming of metal oxide memristive switches," *Nanotechnology*, vol. 20, no. 21, May 2009, Art. no. 215201.

[29] G. Bersuker et al., "Metal oxide resistive memory switching mechanism based on conductive filament properties," *J. Appl. Phys.*, vol. 110, no. 12, Dec. 2011, Art. no. 124518.

[30] S. Wiefels, C. Bengel, N. Kopperberg, K. Zhang, R. Waser, and S. Menzel, "HRS instability in oxide-based bipolar resistive switching cells," *IEEE Trans. Electron Devices*, vol. 67, no. 10, pp. 4208–4215, Oct. 2020.

[31] V. G. Karpov and D. Niraula, "Log-normal statistics in filamentary RRAM devices and related systems," *IEEE Electron Device Lett.*, vol. 38, no. 9, pp. 1240–1243, Sep. 2017.

[32] P. Huang et al., "Analytic model for statistical state instability and retention behaviors of filamentary analog RRAM array and its applications in design of neural network," in *IEDM Tech. Dig.*, Dec. 2018, pp. 40.4.1–40.4.4.

[33] B. Q. Le et al., "RADAR: A fast and energy-efficient programming technique for multiple bits-per-cell RRAM arrays," *IEEE Trans. Electron Devices*, vol. 68, no. 9, pp. 4397–4403, Sep. 2021.

[34] A. Fantini et al., "Intrinsic program instability in HfO_2 RRAM and consequences on program algorithms," in *IEDM Tech. Dig.*, Dec. 2015, pp. 7.5.1–7.5.4.

[35] S. Clima et al., "Intrinsic tailing of resistive states distributions in amorphous HfO_x and TaO_x based resistive random access memories," *IEEE Electron Device Lett.*, vol. 36, no. 8, pp. 769–771, Aug. 2015.

[36] C. Peters, F. Adler, K. Hofmann, and J. Otterstedt, "Reliability of 28 nm embedded RRAM for consumer and industrial products," in *Proc. IEEE Int. Memory Workshop (IMW)*, May 2022, pp. 1–3.

[37] S. Fukuyama, K. Maeda, S. Matsuda, K. Takeuchi, and R. Yasuhara, "Suppression of endurance-stressed data-retention failures of 40 nm TaO_x-based ReRAM," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Mar. 2018, pp. P-MY.4-1–P-MY.4-5.

[38] S. Muraoka, T. Ninomiya, Z. Wei, K. Katayama, R. Yasuhara, and T. Takagi, "Comprehensive understanding of conductive filament characteristics and retention properties for highly reliable ReRAM," in *Proc. Symp. VLSI Technol.*, 2013, pp. T62–T63.

[39] C. Bengel et al., "Reliability aspects of binary vector-matrix-multiplications using ReRAM devices," *Neuromorphic Comput. Eng.*, vol. 2, no. 3, Sep. 2022, Art. no. 034001.

[40] S. Menzel, U. Böttger, M. Wimmer, and M. Salinga, "Physics of the switching kinetics in resistive memories," *Adv. Funct. Mater.*, vol. 25, no. 40, pp. 6306–6325, Oct. 2015.

[41] K. Fleck, C. La Torre, N. Aslam, S. Hoffmann-Eifert, U. Böttger, and S. Menzel, "Uniting gradual and abrupt set processes in resistive switching oxides," *Phys. Rev. Appl.*, vol. 6, no. 6, Dec. 2016, Art. no. 064015.

[42] F. Cüppers et al., "Exploiting the switching dynamics of HfO_2-based ReRAM devices for reliable analog memristive behavior," *APL Mater.*, vol. 7, no. 9, Sep. 2019, Art. no. 091105.

[43] C. Bengel et al., "Variability-aware modeling of filamentary oxide-based bipolar resistive switching cells using SPICE level compact models," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 12, pp. 4618–4630, Dec. 2020.

[44] V. Milo et al., "Accurate program/verify schemes of resistive switching memory (RRAM) for in-memory neural network circuits," *IEEE Trans. Electron Devices*, vol. 68, no. 8, pp. 3832–3837, Aug. 2021.

[45] L. Grenouillet et al., "16 kbit 1T1R O_xRAM arrays embedded in 28 nm FDSOI technology demonstrating low BER, high endurance, and compatibility with core logic transistors," in *Proc. IEEE Int. Memory Workshop (IMW)*, May 2021, pp. 1–4.

[46] H.-X. Zheng et al., "Reducing forming voltage by applying bipolar incremental step pulse programming in a 1T1R structure resistance random access memory," *IEEE Electron Device Lett.*, vol. 39, no. 6, pp. 815–818, Jun. 2018.

[47] X. Xu et al., "First demonstration of OxRRAM integration on 14 nm FinFET platform and scaling potential analysis towards sub-10nm node," in *IEDM Tech. Dig.*, Dec. 2020, pp. 24.3.1–24.3.4.

[48] S. N. Ong et al., "A 22 nm FDSOI technology optimized for RF/mmWave applications," in *Proc. IEEE Radio Freq. Integr. Circuits Symp. (RFIC)*, Jun. 2018, pp. 72–75.

[49] M. Kim and P. Smaragdis, "Bitwise neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1–5.

[50] Z. Chen and J. Gu, "Analysis and design of energy efficient time domain signal processing," in *Proc. Int. Symp. Low Power Electron. Design*, Aug. 2016, pp. 100–105.

[51] M. Gansen et al., "Discrete steps towards approximate computing," in *Proc. 23rd Int. Symp. Quality Electron. Design (ISQED)*, Apr. 2022, pp. 1–6.

[52] M. Wiatr and S. Kolodinski, "22FDX^{TM} technology and add-on-functionalities," in *Proc. 49th Eur. Solid-State Device Res. Conf. (ESSDERC)*, Sep. 2019, pp. 70–73.