

This is the peer reviewed version of the following article:

Ultra-low power logic in memory with commercial grade memristors and FPGA-based smart-IMPLY architecture / Benatti, L; Zanotti, T; Pavan, P; Puglisi, Fm. - In: MICROELECTRONIC ENGINEERING. - ISSN 0167-9317. - 280:(2023), pp. 112062-112072. [10.1016/j.mee.2023.112062]

Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

03/07/2024 16:57

(Article begins on next page)

Ultra-Low Power Logic in Memory with Commercial Grade Memristors and FPGA-Based Smart-IMPLY Architecture

Lorenzo Benatti, Tommaso Zanotti, Paolo Pavan, Francesco Maria Puglisi

Dipartimento di Ingegneria “Enzo Ferrari”, Via P. Vivarelli 10/1, 41125 – Modena (MO) - Italy

Corresponding author email: lorenzo.benatti@unimore.it, francescomaria.puglisi@unimore.it phone: +39-059-2056324

Abstract— Reducing power consumption in nowadays computer technologies represents an increasingly difficult challenge. Conventional computing architectures suffer from the so-called von Neumann bottleneck (VNB), which consists in the continuous need to exchange data and instructions between the memory and the processing unit, leading to significant and apparently unavoidable power consumption. Even the hardware typically employed to run Artificial Intelligence (AI) algorithms, such as Deep Neural Networks (DNN), suffers from this limitation. A change of paradigm is so needed to comply with the ever-increasing demand for ultra-low power, autonomous, and intelligent systems. From this perspective, emerging memristive non-volatile memories are considered a good candidate to lead this technological transition toward the next-generation hardware platforms, enabling the possibility to store and process information in the same place, therefore bypassing the VNB. To evaluate the state of current public-available devices, in this work commercial-grade packaged Self Directed Channel memristors are thoroughly studied to evaluate their performance in the framework of in-memory computing. Specifically, the operating conditions allowing both analog update of the synaptic weight and stable binary switching are identified, along with the associated issues. To this purpose, a dedicated yet prototypical system based on an FPGA control platform is designed and realized. Then, it is exploited to fully characterize the performance in terms of power consumption of an innovative Smart IMPLY (SIMPLY) Logic-in-Memory (LiM) computing framework that allows reliable in-memory computation of classical Boolean operations. The projection of these results to the nanoseconds regime leads to an estimation of the real potential of this computing paradigm. Although not investigated in this work, the presented platform can also be exploited to test memristor-based SNN and Binarized DNNs (i.e., BNN), that can be combined with LiM to provide the heterogeneous flexible architecture envisioned as the long-term goal for ubiquitous and pervasive AI.

Keywords – Memristor, Self-Directed Channel, FPGA, Low-Power Computing, Smart ImPLY.

1. INTRODUCTION

During the last five decades, the microelectronics industry has been continuously evolving thanks to Moore’s law predicting an exponential increase in the number of transistors per chip. Since the 1970’s, in fact, the advances achieved in computing technology have been driven by the miniaturization of metal-oxide-semiconductor field effect transistor (MOSFET), doubling the number of integrated transistors in a microprocessor chip approximately every two years [1]. This exponential increase of device density on chips allowed a continuous gain in computational performances and

led to the development of today’s digital complementary metal-oxide-semiconductor (CMOS) microprocessors. This progress also allowed a drastic drop in costs, rapidly stimulating the diffusion of new technologies and semiconductor devices improving all fields in electronics. However, fundamental issues have recently slowed down these trends, affecting the efficiency increase. Beyond scaling [2] and thermal [3] problems, another fundamental and challenging obstacle for the Moore’s law is known as memory wall [4]. In fact, in conventional processors, the central processing unit (CPU) handles operations at a much higher speed than that needed to access the memory where the data are stored, causing a severe performance bottleneck [5]. The cause of this central issue is the physical separation of CPU and memory in the von Neumann architecture of current digital computers, hence the name von Neumann bottleneck (VNB).

Recently, these limitations have pushed the research towards innovative architectures and approaches, leading to the investigation of novel concepts at the device [6]–[8], circuit [9], [10], and system level [11], [12], with results such as hybrid memory-logic integration for a better memory storage, brain-inspired computing [13]–[15] and in-memory computing [16]–[18]. This transition towards a more efficient computing technology is pushed by a new class of emerging non-volatile memories (NVMs), that are currently undergoing development and are being actively researched by industries and universities with the aim of revolutionizing the existing memory hierarchy [19]. Among them, a particular device that is increasingly gaining interest in the development of these new paradigms is the memristor [20] (such as Resistive Random Access Memory (RRAM) [21], [22]), a 2-terminal NVM considered an optimum candidate to work both in ultra-low power analog and digital Neural Networks (NN) [23], [24], bio-inspired architectures for associative learning [25]–[27], and Logic-in-Memory (LiM) circuits [28]–[30]. Although the requirements for implementing digital LiM architectures are far more attainable with respect to Neural Networks (NN) [31], the experimental validation of this approach is still poorly investigated, especially with commercial-grade RRAMs or memristors. In fact, evaluating the performances of memristors available on the public market can be seen as an indicator of the actual large scale implementation readiness of the investigated approach.

In this work, we investigate the suitability of commercially available memristors for both analog NN and digital LiM applications evaluating, for the latter, the achievable

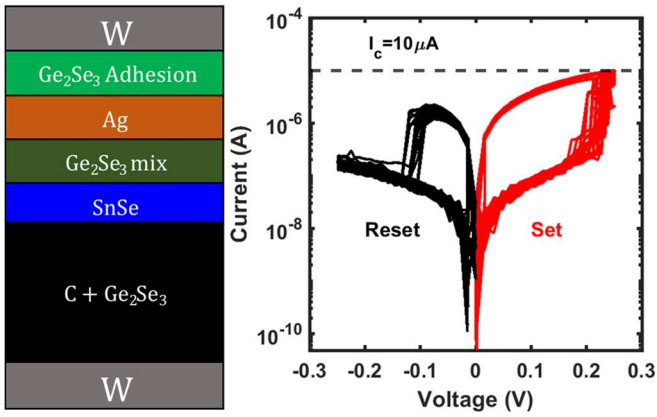


Fig. 1 – (left) Layout of SDC memristor. (right) I-V curve (sweep rate = 312 mV/s) for 50 cycles of set ($V > 0$) and reset ($V < 0$), using a current compliance (I_c) of 10 μ A.

performance and energy efficiency. Characterizations results are used to design and develop a prototypical FPGA-based platform allowing to correctly drive a memristor array for the execution of digital LiM operations. The platform is then exploited to evaluate the energy efficiency performances of the innovative Smart-IMPLY (SIMPLY) computing framework [32], a recently introduced LiM paradigm, which are then compared with those of the traditional CMOS counterpart. Although implemented with commercial grade memristors, the same methodology is applicable to other memristive technologies, such as RRAMs or phase change memories (PCM) [33].

The paper is organized as follows: Section 2 describes the layout and the physical mechanism of the commercial-grade Self-Directed Channel memristors, to our knowledge the only available on the market, and their characterization results for analog and binary LiM behavior. In Section 3, the SIMPLY LiM architecture advantages over the classic IMPLY are presented. In Section 4, we analyze the details of the designed and realized FPGA-based platform, and the performed experiments. Finally, the energy consumption profiles extracted from these tests are then presented in Section 4 and discussed in Section 5, comparing the performances of a 32-bits full adder (FA) vs. its traditional CMOS counterpart.

2. MATERIALS AND METHODS

2.1 Self-Directed Channel Memristors

The memristors employed in this work, called Self-Directed Channel (SDC), are developed by Knowm Inc. [34]. We here characterize them and verify their potential in neural networks and logic applications.

Differently from metal-oxide RRAMs [21], which rely on the formation and dissolution of a conductive filament for the switching mechanism, these SDC memristors are ion-conducting devices that change their resistance due to the movement of Ag^+ ions into the device structure [35]. Figure 1 (left) shows the stack layout of the studied devices. Despite the high number of thin layers, the fabrication mechanism is simple and reliable. In fact, the deposition of all layers, including the top electrode, is done in-situ in one processing step by means of sputtering. The constant separation between the Ag-source, consisting in the $Ge_2Se_3/Ag/Ge_2Se_3$ layers, and the top electrode, allows also high temperature processes and operations, including long-term continuous operation at 150

$^{\circ}C$ [35]. Also, no high voltage forming step is required, meaning that the same set voltage required during the normal device operation can be used to switch a pristine device into a low resistance state [36]. Furthermore, the required programming voltages and compliance current values are considerably lower with respect to classical metal-oxide RRAMs [21], with a consequent decrease in power consumption.

Each package is constituted by eight discrete SDC devices that are initially in a high resistance state ($M\Omega - G\Omega$ range) [37]. The first set operation generates Sn ions from the SnSe layer and forces them into the active Ge_2Se_3 layer. Sn ions, in fact, are expected to facilitate the incorporation of Ag into the active layer at the Ge-Ge bonding sites [38]. This occurs through an energetically favorable process in which the electrons entering the active layer from the negative bottom electrode, concurrently with the formation of Sn ions from the SnSe layer, enable formation of a pair of self-trapped electrons in the Ge_2Se_3 active layer strongly localized around the Ge-Ge dimers present in this Ge-rich glass [38]. This results in the distortion of the Ge-Ge bond by means of the reaction with Ag, creating an ‘opening’ near the Ge-Ge sites, providing the access for Ag^+ and creating a natural conductive self-directed channel within the active layer for the movement of Ag^+ during device operation, since Ag has a tendency to agglomerate with other Ag atoms. This pathway does not consist in a conductive metallic filament between the two electrodes, but it is simply a channel with a resistance imposed by the varying concentrations of Ag within it. The resistance is tunable in the lower and higher directions by movement of Ag onto or away from these agglomeration sites through application of either a positive (set) or negative (reset) potential, respectively, across the device [36]. These devices have been characterized using a Keithley 4200-SCS system. Figure 1 (right) depicts the DC switching process. The absence of a real conductive filament determines a set transition (resulting in a low resistive state, LRS) that is somewhat less abrupt with respect to the one observed in metal-oxide technology [21]. On the contrary, reset (resulting in a high resistive state, HRS) occurs in a sudden way, which breaks the loop symmetry. The low power consumption is given since the currents are small and voltage levels required for set and reset are notably low, since there is no need to initiate a real soft breakdown in the device.

2.2 Analog Behaviour

The potential of analog memristor-based NN, taking inspiration from biological brain mechanisms, relies on the ability to actively strengthen (potentiation) or inhibit (depression) the analog synaptic weights of the network, depending on the neuron’s activity [39]–[41]. This results in an extremely energy efficient solution to overcome the VNB limitations. However, the requirements in terms of the number of stable levels achieved by means of potentiation and depression spans from 64 (SNN) to 100 (DNN) [31]. To verify if a satisfactory retention of analogue values is achievable with this technology, a sequence of negative (depression) and positive (potentiation) voltages with same amplitude and different time widths has been applied (Fig. 2a). Experiments show that depression pulses with small width contribute to a resistance increase only in devices at low resistance levels. Gradually increasing the pulse width allows to push the

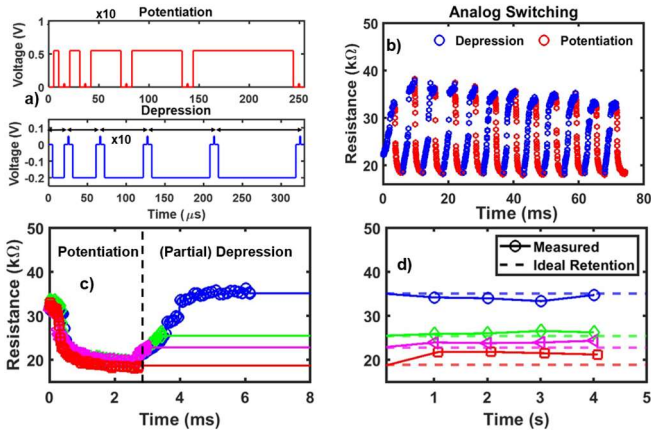


Fig. 2 – a) Sequences for potentiation and depression. Each pulse/read segment is actually repeated 10 times before changing pulse width. b) Analog switching obtained by repeatedly applying the sequence in a). c) Retention drifts observed by reading the device state after full potentiation (red), partial depression (magenta, green), and full depression (blue). d) Few seconds are sufficient to have significant drifts from the programmed state (dotted lines), especially when starting from low resistive values.

memristor to higher resistance values, until a saturation level is reached (depending on the pulse train amplitude and width). On the contrary, potentiation pulses with modest widths affect the device resistive state only if the latter is high enough. Figure 2b shows the measured trends, revealing the possibility to obtain acceptably smooth depression and potentiation mechanisms with these devices. However, the ambitious requirements for neuromorphic applications seem far from being met. First, the re-iterated application of the same sequence does not result in the same resistance change every time, causing an important reproducibility problem. In fact, although the potentiation process leads always to the same final resistance value of about 20 k Ω (i.e., the bottom of the dynamic range in Fig. 2b), the depression sequence output presents a larger variability (i.e., the top of the dynamic range in Fig. 2b is less stable). Furthermore, the analogue switching mechanism is not smooth enough, and so insufficient to guarantee an appropriate number of stable and separated levels for a real neuromorphic application. A higher number of levels can, in principle, be obtained by enlarging the resistance window by acting, for example, on the depression sequence, with experiments suggesting that a pulse amplitude increase is more effective than a pulse width increase (not shown).

Finally, as actually expected from this technology documentation [40], the state’s retention is insufficient for reliable analog applications. In fact, for each state the resistance value drifts after few seconds (Fig. 2c-d), exhibiting larger effects when the device is read in low resistive states, which makes it unlikely to attain 64 stable levels (i.e., the standard requirement for SNNs [31]).

2.3 Binary Behaviour

Although a neuromorphic approach is possible, it is still far from being dependably viable with these devices. Thus, a memristor binary behavior with satisfactory reliability, combined with an in-memory computing paradigm, will ensure a first step in the direction of a new generation of ultra-low power computing architectures. The requirements needed for a complete substitution of the classical memory hierarchy are, actually, extremely more attainable with respect to neural

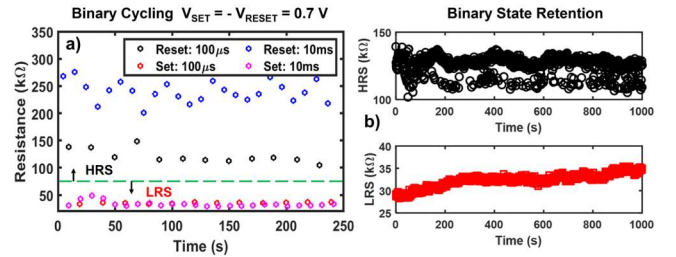


Fig. 3 – a) HRS and LRS cycling study with different set-reset PWs (100 μ s and 10ms), using the same $V_{SET} = -V_{RESET} = 0.7$ V. Resistance values are read using a 50 mV 10 ms pulse with a delay of 10 s ($t_{SET} = t_{RESET} = 100$ μ s) and 5 s ($t_{SET} = t_{RESET} = 10$ ms) from each set (reset) switch to let vanish possible transient fluctuations. b) Binary state retention investigation. A satisfactory LRS and HRS retention is found for 1000 s experiments.

networks necessities [31]. In fact, NN applications ideally require the device analog programming with linear resistance updates which demand much more complex circuits and device programming sequences, with respect to applications based on binary storage devices.

Figure 3a shows that a satisfactory and repeatable window can be obtained by using relatively fast voltage pulses and by using higher voltages than those strictly needed to allow quasi-static operation (i.e., DC set and reset voltages). Experimental data are reported for different set and reset pulse widths (PWs, 100 μ s and 10 ms) and read after an arbitrary delay of few seconds (10 s and 5 s) to exclude possible transient effects as those characterizing analogue switching. To obtain a reasonable retention the compliance current has to be pushed nearly at its limits, which in these devices is nominally 50 μ A, in order to stabilize the created channel and minimize the short-term plasticity effect [42]. The risk is in fact that after a set or reset operation, the reached resistance value abruptly drifts to an intermediate value between HRS and LRS, as explicitly reported in [42] and also seen in this section. In these conditions, devices exhibit a satisfactory HRS and LRS retention for at least 1000 s, as reported in Fig. 3b. However, this technology is already reported to not guarantee a stable state retention beyond 30 minutes [43], [44]. Nonetheless, the available retention displayed by the commercial SDC devices when used as binary elements, see Fig. 3, is sufficient for the demonstration of the advantages introduced by LiM computing paradigms and of the usefulness of the proposed FPGA-based platform as a tool for studying and benchmarking the performance of such innovative computing paradigms.

3. THEORY

3.1 Smart-Imply Advantages

Memristors allow the execution of the material implication (IMPLY) and FALSE operations, that in recent years gained worldwide interest [45] as they form a complete logic group, i.e., implemented in an appropriate circuit, IMPLY-FALSE sequences can be used to successfully compute any logic operation [46]. Discarded for their incompatibility with MOSFET designs, these operations exploit the stateful property of memristors [47], meaning that a memristor device functions both as a logic gate and as a memory element, enabling a real LiM approach. Specifically, logic-0 is represented by a device in HRS, logic-1 by a device in LRS.

The FALSE operation consists in restoring the HRS state of a single device, by means of a negative voltage pulse, called

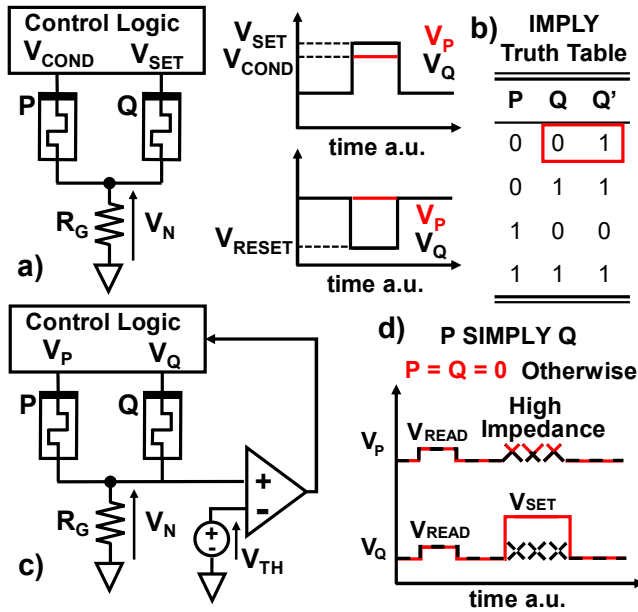


Fig. 4 – a) IMPLY working principle and relative truth table b). In b) the red box evidences the fact that the input $P=Q=0$ is the only exhibiting $Q \neq Q'$. This leads to the SIMPLY architecture c), that perform a set operation on Q only if P-read-Q sense $V_N < V_{TH}$ d) its working principles.

V_{RESET} , independently from the initial logic state of the device. To achieve the logical operation of P-IMPLY-Q, which corresponds to (NOT P) OR Q, two positive voltages are applied simultaneously to devices P and Q (as shown in Fig. 4a). The resulting resistive value of Q, called Q' , obtained from this operation, represents the output. It is crucial to appropriately determine the values of the positive voltages, referred to as V_{COND} and V_{SET} , in order to ensure that the logic state of P remains unchanged and to obtain the correct truth table for the operation (as illustrated in Fig. 4b). It has also been proved that a sequence of IMPLY and FALSE operations in an array of a specific number of memristors (the number of inputs plus at least 2 additional memristors [48]), enables the actual in-circuit computation of any logic operation [46], with promising results in term of energy efficiency and integration density projections. However, the critical choice of V_{COND} and V_{SET} imposes an accurate preliminary phase before designing a fully functional IMPLY gate. Even if its functionality has been explored, simulated [49]–[51] and experimentally demonstrated [47] in many works, not all the issues have been taken into account. Although promising, this scheme, in fact, is also affected by memristors severe limitations such as the logic state degradation [32], [49], [50], [52], cycle to cycle resistive variability and random telegraph noise (RTN) [49], [50], [53], which can prevent the correct circuit functionality in long-term operations if not considered during the design phase. To solve the issues observed in the traditional material implication scheme, a novel Smart-IMPLY (SIMPLY) LiM scheme has been developed [32].

The idea behind SIMPLY starts from the observation of the IMPLY truth table. During P-IMPLY-Q operation, the state of Q changes only when the input combination is $P = Q = 0$, otherwise Q retains its initial state, Fig. 4b. So, it is possible to distinguish this input combination from all the others by applying two simultaneous small read voltage pulses at the top electrodes (TEs) of P and Q and comparing the voltage at node N (V_N) with a predefined threshold voltage V_{TH} (Fig. 4c). The

range of all possible V_{TH} values that allow a correct operation is called read margin (RM), that must be sufficiently large to have a clear distinction of the $P = Q = 0$ case from the others. This value can be improved by applying higher V_{READ} or more negative V_{FALSE} (to increase the HRS/LRS ratio [54]), but with a consequent increase in power consumption. The output of the comparator, which must have a small footprint and dissipate low energy [54], is then fed back to a control logic with analog tri-state buffers. This control circuit applies a V_{SET} pulse to the TE of Q if $V_N < V_{TH}$ or keeps the P and Q TEs at high impedance, as shown in Fig. 4d. The application of only a small read voltage in three out of four cases of the IMPLY truth table results also in intriguing energy savings with respect to the traditional implementation of the IMPLY [54], with a consequent improved efficiency in calculating logic operations that require long IMPLY-FALSE sequences. Therefore, this strategy totally removes the trade-off deriving from the challenging choice of V_{COND} and V_{SET} .

4. RESULTS

4.1 FPGA-Based Platform

The promising obtained results encouraged the development of an evaluation board able to control an array of memristors to realize the core operations of both the IMPLY and the SIMPLY logic schemes, with the aim of measuring the energy consumption associated with these logic operations and their sequences, within the possibility of the FPGA specifications in terms of PWs. Results are then projected to the nanosecond regime [55] to estimate the attainable performance of this ultra-low power approach when run at GHz speed in fully-integrated Very Large Scale of Integration (VLSI) chips.

The entire system is composed by a DE1-SoC 5CSEMA31C6F (Cyclon V) FPGA, a custom digital-to-analog interface (DAI), and an array of 8 Knowm memristors (Fig. 5). The FPGA adoption provides the possibility to instantiate the desired hardware, programmed using VHDL and Verilog languages, hence resulting in a clever exploitation of the system resources. In particular, for this work the FPGA platform employs only 2 % of logic utilization (601/32070), expressed in Adaptive Logic Module (ALM), the number of used registers is 703, the number of pins is 87/457 (19 %), and the number of DSP blocks (containing shift registers) is 4/87 (5 %). DE1-SoC 5CSEMA31C6F (Cyclon V) FPGA implements an internal clock signal at 50MHz, enabling the creation of multiple timing signals for each virtual block. The internal ADC features 8 channels with a 12-bits resolution, input voltage allowed in the range of 0 - 4.096 V, with a sample read frequency up to 40 MHz. Only one channel is needed in our case, which receives the feedback signal V_N that decides, after the comparison with a specifically sized V_{TH} , the next driving voltages to be used in the following SIMPLY operation (Figs. 4c, 5). This voltage value is converted in a decimal base and displayed on the 7-segment digital display for quick manual inspection. The desired platform must be able to handle simultaneously n ($1 \leq n \leq 8$) memristors and provide different PWs and amplitudes for the required TE signals (V_{READ} , V_{SET} , V_{RESET} , V_{COND}). Three different operation modes can be outlined: i) Operations on a single specific device, fundamental for FALSE operations, requiring the system to ensure also negative pulses. It is also crucial for

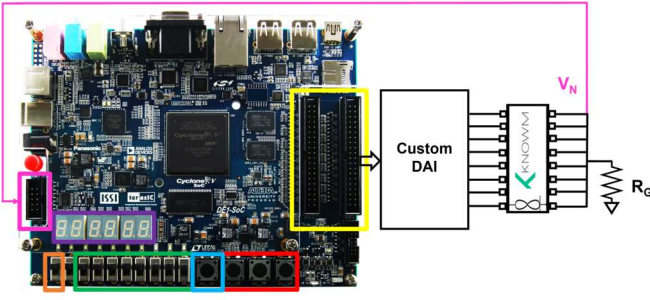


Fig. 5 – Schematic of the FPGA-based platform, composed by the FPGA, the developed custom digital to analog interface (DAI) and the array of packaged SDC Knownm memristors. V_N (magenta) is read by the FPGA ADC (magenta box), while the custom DAI is fed by the GPIO pins (yellow box). The red buttons allow to apply single operations (read, set, reset) to the memristors selected by the green switches, while the blue button starts a pre-coded sequence on specific devices. Orange switch supports the IMPLY operation by enabling the presence of a V_{COND} different from V_{SET} .

SIMPLY, that provides a set pulse when the $P = Q = 0$ case is found. *ii*) Operations on two devices. IMPLY and SIMPLY require the application of simultaneous driving voltages on two different devices. IMPLY requires also the possibility to apply different amplitude pulses (V_{SET} and V_{COND}) to the two different devices, while SIMPLY requires applying the same voltage, V_{READ} , to both devices. *iii*) Operations on n devices. Complex sequences can benefit from the execution of the SIMPLY operation with more than two inputs [56]. A simultaneous read pulse on more than two memristors must be applied in this case. Although in this work we do not explore this possibility, the platform is conceived to allow this operation mode for future studies. All these modalities can be performed ‘manually’ or in a pre-defined sequence guided by the specific FPGA code. The employed FPGA provides, in fact, ten switches and four buttons, each linked to a specific function. Eight switches are used as memristors selectors, initializing the needed enabling signals to drive the correct devices. The buttons array activates the fundamental functions. Three of them are dedicated to start the read, set, and reset operations on the memristors selected by the switches. In this way, each device can be read, driven in HRS or LRS independently from a specific instruction sequence. The fourth button launches a coded sequence (SIMPLY-based or IMPLY-based) on specific devices as specified in the code itself. Another switch ($V_{COND-ENABLE}$) allows to enable the presence of a positive pulse, different from V_{SET} , during IMPLY. The PWs are programmed by means of a Pulse Width Modulation (PWM) VHDL component. The FPGA General Purpose Input/Output (GPIO), composed by 40 pins, supports only digital signals, consisting in a logic-0 of 0 V and a logic-1 of 3.3 V. The programmed PWM in the FPGA logic can only handle the output signal pulse width and not its amplitude, stressing the necessity for a proper conditioning circuit. Therefore, we implemented a custom digital to analog interface (DAI) that allows to transfer the correct input voltages to the respective memristors (Fig. 6). It is composed by: *i*) 2 digital potentiometers (MCP41010), ranging from 0 to 10 k Ω ; *ii*) 3 analog tri-state buffers (TS12A12511); *iii*) 8 4-channel analog multiplexers (TMUX6104); *iv*) 2 operational amplifiers; *v*) 1 2-inputs OR; *vi*) 1 n-MOSFET; *vii*) 7 resistors.

The memristors are selected by means of an array of multiplexers. Each multiplexer is connected to the TE of a single memristor and is activated through a digital enable pin

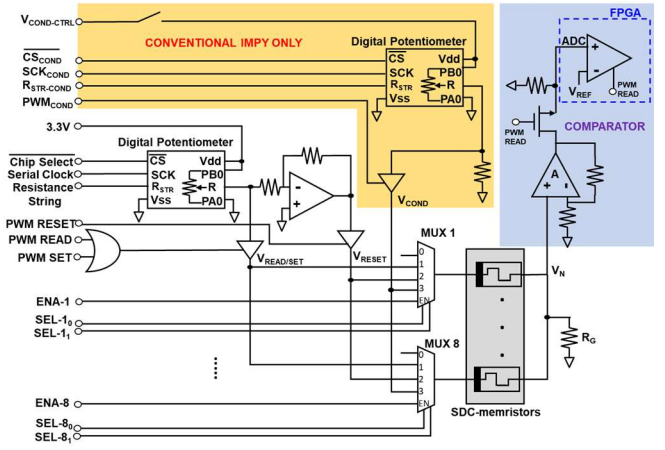


Fig. 6 – Schematic of the custom DAI connected to the memristors array and to the ADC internal to the FPGA. In yellow the circuit dedicated to implementing V_{COND} (only for conventional IMPLY). In blue the comparator circuit, where the signal is amplified by a factor $A = 4.3$ to improve its readability by the ADC inside the FPGA. An n-MOS is inserted between the op-amp and the ADC to avoid negative pulses inside the FPGA during the reset operation, which could damage the platform.

(ENA1, ENA2, ..., ENA8, see Fig. 5). When enabled, the output of the multiplexer is driven depending on the desired operation (read, set or reset), which is determined by the state of the selector bits (sel1-0, sel1-1, sel2-0, sel2-1, ..., sel8-0, sel8-1, see Fig. 6). When a memristor is not selected, its TE is kept floating by the tri-state I/O pins of the FPGA. The output voltages are adjusted using programmable digital potentiometers in a voltage divider configuration (see Fig. 6). One digital potentiometer is used to generate both V_{SET} and V_{READ} , while the other is used to generate V_{COND} (yellow area). V_{RESET} is also determined by the gain of the associated operational amplifier in inverting configuration, which receives V_{SET} as an input. For each read, set, reset operation the respective enabling PWM signals activate an analog tri-state buffer only for a determined period of time, which allows the signal transmission to the multiplexer array, sending the pulse only at the selected memristors. For the SIMPLY sequence, an amplifier is used to boost the signal before the comparator, that is implemented directly inside the FPGA by means of the ADC, thus improving the signal to noise ratio (blue area). The comparison obviously is done only during the read operation. To avoid the possibility of the ADC input experiencing negative transient voltage during the reset operation, a n-MOSFET and a resistor are placed before the ADC pin to realize a pull-down action.

Although this FPGA-based platform has been designed for implementing both IMPLY and SIMPLY core operations, the high voltage amplitudes (0.7V, Fig. 3-7b) needed to accomplish a satisfactory retention on the binary states in these SDC-memristors discourage the demonstration of the traditional IMPLY strategy. In fact, the large difference between the quasi-static (0.2 V) and pulsed (0.7 V) set voltage values suggests the impossibility to find an adequate V_{COND} - V_{SET} pair. Indeed, since these devices can quickly switch from the HRS to the LRS even when subject to voltage pulses with amplitude lower than 0.7 V, but without a sufficient state retention, it is impossible to find a V_{COND} value, smaller than but still close to V_{SET} , that does not impair the correct circuit operation, and likely any V_{COND} - V_{SET} pair will cause an unwanted state change in P during P-IMPLY-Q. The SIMPLY approach, that totally removes this delicate trade-off, is

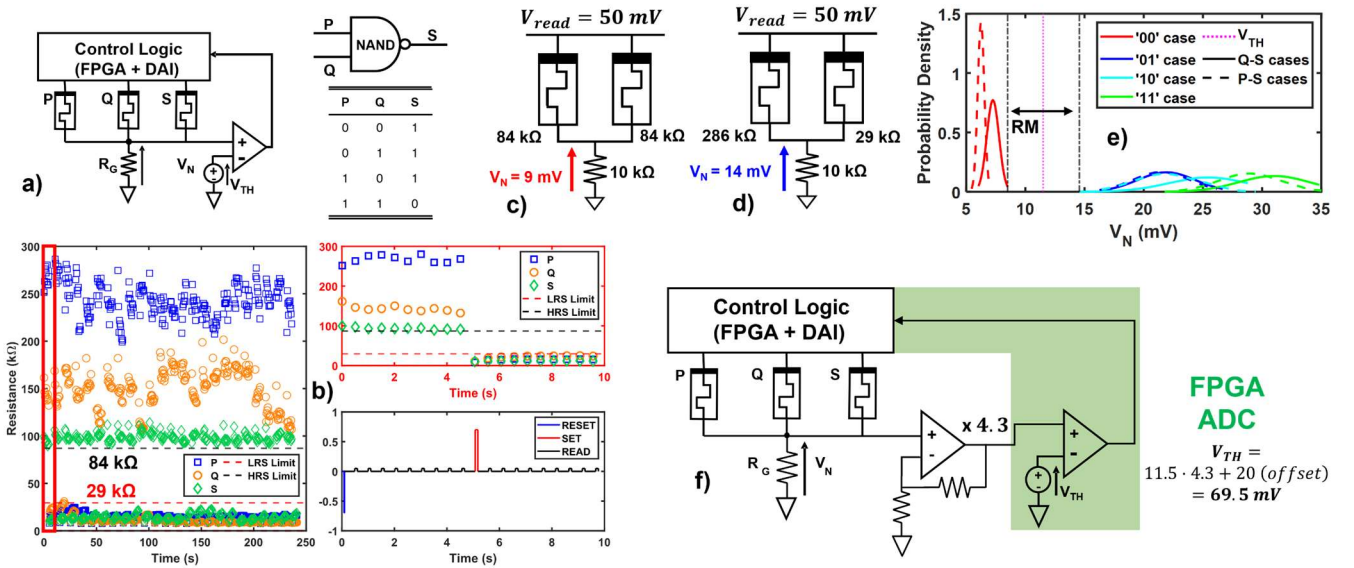


Fig. 7 – a) SIMPLY architecture for NAND logic gate and relative truth table. b) Characterization of cycling and stability (short-term retention) of employed devices, showing acceptable HRS/LRS windows. Set (reset) is obtained applying pulses of ± 0.7 V and $PW = 10$ ms, followed by a train of 9 read pulses (50 mV 10 ms) to analyze the state retention, as shown in the red box. c-d) Worst ‘00’ and ‘01’ case and relative V_{NS} , obtained considering c) the most unlikely ‘00’ case with two different memristors with the lowest possible HRS ($V_N = 9$ mV) and d) ‘01’ (or ‘10’) combination with highest LRS and HRS ($V_N = 14$ mV). V_{TH} should then be in the [9 14] mV range. e) Probability distribution of V_N considering operations between Q-S (solid lines) and P-S (dotted lines) devices, evidencing the presence of a satisfactory RM. The derived V_{TH} (11.5mV) is then used in f) to choose the gain of the amplifier in order to facilitate V_N readout at the ADC input inside the FPGA. Then, considering the amplification and FPGA ADC offset, $V_{TH} = 11.5 \text{ mV} \cdot 4.3 + 20 \text{ mV}(\text{offset}) = 69.5 \text{ mV}$.

therefore a better solution for implementing fully functional LiM circuits with these devices. Furthermore, although neither implemented nor investigated because of the analog switching instability, the developed platform can also be exploited to test the core operations of memristor-based NNs, such as analog spiking NN (SNN) [57]–[59] or binary NN (BNN) [60]–[63]. In fact, in SNNs, memristors implement the analog synapses of the artificial neurons, and synaptic potentiation and depression rules could be evaluated on the developed platform thanks to the possibility to dynamically adjust the PWs. In BNNs, since neurons’ weights and activations are binary, all the core operations are logic operations that can be implemented with memristor based LiM paradigms, such as SIMPLY [64]. The remaining FPGA switch (Fig. 5) could also be adopted to select if the desired switching mechanisms should be analog of digital, enabling in the same platform both implementations.

Although not currently tested on the present prototype, also other LiM computing paradigms could be tested on the proposed platform. For instance, scouting logic [65], which follows a similar principle as the one used for SIMPLY could be implemented with minor changes to the FPGA code, while the platform could also implement the Memristor-Aided Logic (MAGIC) [66] (i.e., another stateful LiM paradigm) by substituting the R_G resistor (see Fig. 6) with an n-MOSFET.

4.2 Performed Experiments: NAND

The circuit functionality is demonstrated by correctly implementing and running a NAND function, a simple yet prototypical logic function that can be implemented in the SIMPLY architecture on an array of three memristors (connected to ground through a common resistor R_G), Fig. 7a, and three FALSE/IMPLY logic operations.

The circuit functionality and its performance in terms of energy consumption are evaluated by adopting pulses with different durations (down to the limitations imposed by the FPGA platform). NAND appears so to be the perfect

candidate for evaluating, in a simple way, the functionality of the developed architecture through the execution of a complete logic function. The demonstration of the NAND truth table permits also to measure the performances, in terms of energy consumption, of the single read, set and reset operations.

Considering three memristors (P, Q, S), where S is the device in which the result of the logic function is stored, and P and Q are the devices that host the input bits, the NAND function can be executed by means of the following sequence: i) FALSE (= reset) S, ii) SIMPLY (P, S), iii) SIMPLY (Q, S). As a preliminary step, before starting to evaluate the feasibility and performance of the NAND logic function as implemented in SIMPLY architecture, the selected devices must guarantee a sufficient HRS/LRS ratio in order to correctly work (i.e., being able to distinguish the case in which both input bits are at logic-0 from the other cases). Figure 7b shows that, although the LRS/HRS window is not particularly large, the chosen memristors satisfy this basic requirement with an acceptable retention. The correct behavior of the proposed circuitual solution, experimentally demonstrated and reported in the following section, reveals that SIMPLY is a reliable architecture that can withstand the non-idealities of memristor devices. A crucial part in designing a SIMPLY architecture relies in sizing the threshold voltage V_{TH} , fundamental for detecting a condition where the involved devices are both in HRS. Figures 7c-d-e show how this value can be calculated observing the statistical variability of the employed memristors, studying the HRS and LRS combination distributions and taking the limit values to decide the optimal V_{TH} . During SIMPLY, the two involved devices are recognized as logic-0 only if, during the read operation ($V_{READ} = 50$ mV), the voltage measured at the common resistor (V_N) is low enough to demonstrate that the examined memristors are both in HRS (henceforth this condition will be called ‘00’). In our practical case, the lowest registered HRS value in these devices is 84 k Ω and leads to a hypothetical

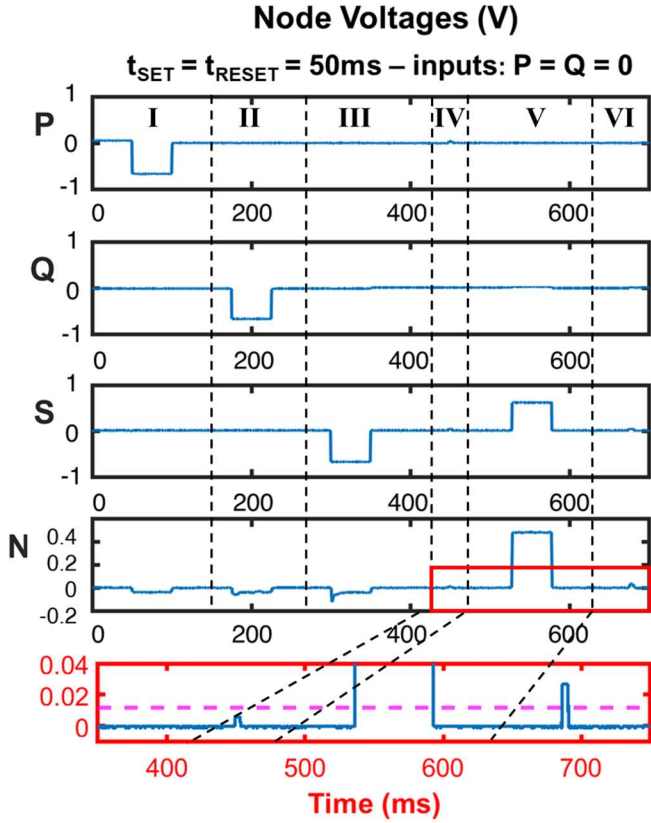


Fig. 8 – Example of oscilloscope voltage traces for each node in a NAND SIMPLY sequence with $P = Q = 0$, highlighting each operation with a time slot (I to VI). I-II slots show the initialization of input RRAMs (in this case $P = Q = 0$, so a reset is applied to both devices). The SIMPLY sequence is then evidenced in III-VI. III indicates the first RESET(S), while IV-V show the first P-SIMPLY-S. After sensing $V_N < V_{TH}$ a set is applied to S. VI shows Q-SIMPLY-S, where $V_N > V_{TH}$. These two operations (red box) are zoomed below to better evidence the difference between V_N at different readings.

limit case for '00' detection characterized by two different memristors both at $84\text{ k}\Omega$ (Fig. 7c, even if such combination is never practically observed in the results in Fig. 7b), resulting in $V_N = 9\text{ mV}$. In all other cases, at least one device is in LRS ('01', '10' or '11'). The worst hypothetical '01' or '10' case is when both devices exhibit the highest LRS and HRS, $29\text{ k}\Omega$ and $286\text{ k}\Omega$, then $V_N = 14\text{ mV}$. Each '11' combination will lead to a higher V_N . Since the read margin (RM) is calculated as the difference between the values of V_N of the worst '00' case and the worst case different from '00', V_{TH} can now be fixed as the mean value of this window (11.5 mV), as shown in Fig. 7e. Now, when V_N is sensed smaller than V_{TH} during reading, the platform executes a set operation on S. However, the actual distribution of measured resistances for all different combinations (Fig. 7e) shows that the actual RM is slightly larger than the one reported above (i.e., $[8.5\text{ } 14.5]\text{ mV}$), bringing more confidence in fixing $V_{TH} = 11.5\text{ mV}$. In order to facilitate the reading at the FPGA ADC side, the small V_N is amplified with a gain of 4.3 (Fig. 7f) by means of an operational amplifier in non-inverting configuration. By considering the additional ADC offset, measured to be 20 mV , leads then to fixing an actual V_{TH} of 69.5 mV for the comparator.

Each NAND operation is obtained by applying always the same amplitude pulses ($V_{SET} = 0.7\text{ V}$, $V_{RESET} = -0.7\text{ V}$, $V_{READ} = 50\text{ mV}$), using the same PW for set and reset operations. For the read operation the applied width is ten times lower than the respective for the set and reset. All the reported widths, if

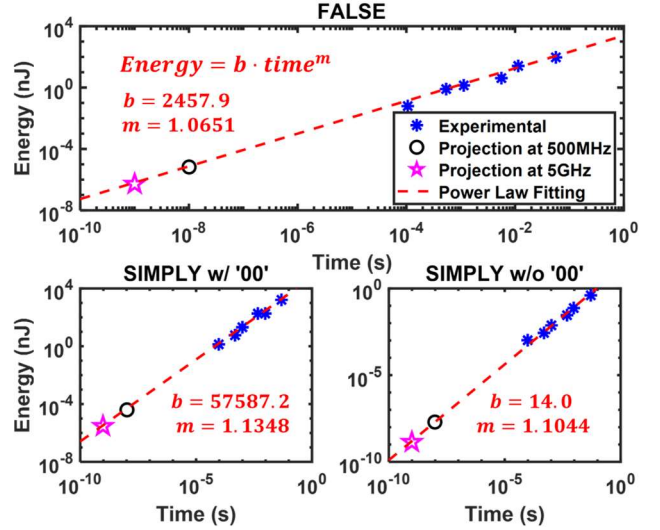


Fig. 9 – Measured performances of single SIMPLY operations (blue asterisks) and their projection at common processors frequencies (magenta star for 5 GHz and black circle for 500 MHz).

	Energy (fJ)	
	500 MHz	5 GHz
FALSE	7.4	0.64
SIMPLY W/ SET	30.8	2.1
SIMPLY W/O SET	0.02	0.002

Tab. I – Energy consumption for single SIMPLY instructions extrapolated at 5 GHz and 500 MHz .

not differently specified, are referred to set and reset operations. The data obtained by scaling down the PWs allowed the projection of the energy consumption of the involved operations. The employed PWs are 50 ms , 10 ms , 5 ms , 1 ms , $500\text{ }\mu\text{s}$, $100\text{ }\mu\text{s}$.

Figure 8 shows how to interpret the experimental results, reporting the oscilloscope voltage traces for each device during the different sequence steps, divided in exemplificative time slots (I-VI). The different graphs show the voltage applied at TE of P, Q, and S devices and the one measured at node N, respectively. Firstly, devices P and Q are initialized with set or reset depending on the desired input configuration (in this case $P = Q = 0$, so a reset is applied to both devices (I-II)). After that, the NAND sequence (III-VI) is performed. The first preliminary (simultaneous) read on P and S senses a '00' case ($V_N < V_{TH}$), and then a set is applied on S (V). The following preliminary (simultaneous) read on Q and S does not sense a '00' case (VI), so the sequence stops with a logic-1 stored in S.

5. DISCUSSION

5.1 Energy Consumption

The execution of NAND operation for each input configuration and at different PWs allows a complete investigation of the performances of this architecture. For each case, the energy consumptions are calculated as follow employing the oscilloscope traces:

$$E_{FALSE} = \int_0^{t_{reset}} V_{FALSE}(t) \frac{V_N(t)}{10\text{ k}\Omega} dt$$

$$E_{SET} = \int_0^{t_{set}} V_{SET}(t) \frac{V_N(t)}{10\text{ k}\Omega} dt$$

$$E_{READ} = \int_0^{t_{read}} V_{READ}(t) \frac{V_N(t)}{10\text{ k}\Omega} dt$$

32-BITS Full Adder Comparison						
	500MHz SIMPLY	5GHz SIMPLY	CMOS W/ VNB*	CMOS W/O VNB*	Other LiM (Exp.)**	Other LiM (Sim.)***
Energy (pJ)	9.3	0.702	132000	0.3-10	0.624	96 - 6464
Delay (ns)	290	29	4000	0.02-1.6	$1.7 \cdot 10^6$	544 - $1.2 \cdot 10^5$
EDP (Js)	$2.7 \cdot 10^{-18}$	$2.0 \cdot 10^{-19}$	$52.8 \cdot 10^{-12}$	$6 \cdot 10^{-24}$ - $2.7 \cdot 10^{-18}$	$1.08 \cdot 10^{-12}$	$5 \cdot 10^{-17}$ - $8 \cdot 10^{-13}$
PDP (Ws)	$9.3 \cdot 10^{-12}$	$6.9 \cdot 10^{-12}$	$1.3 \cdot 10^{-5}$	$1.7 \cdot 10^{-9}$ - $3.4 \cdot 10^{-13}$	$6.4 \cdot 10^{-10}$	$7 \cdot 10^{-9}$ - $9 \cdot 10^{-11}$

Tab. II – Energy consumption, time delay, EDP and PDP comparison among 32-bits FA implemented with SIMPLY at 5GHz and 500MHz and the relative CMOS counterpart (W/ VNB) [70] even considering the ideal case with no energy needed to transmit data from the memory to processing unit and backward (W/O VNB, *[68]–[70]) and other LiM implementations. ** Projections based on the experimental results from [71]. *** Projections based on the simulation results from [54], [56], [72], [73].

From the estimation of the performances of each truth table case of the NAND operation, the single operations energy consumptions can be projected to modern processor speed, impossible to obtain with our platform based on discrete components. The tested system, in fact, is affected by the employed devices and circuit speed limitations. Packaged memristors cannot withstand switching times on the order of nanoseconds because of the packaging parasitic, but high speed can be reached in integrated circuits. The FPGA ADC maximum speed of 40 MHz also does not allow to scale the PW any further. Since the read time is ten times less than the set (reset) time, the minimum tested read pulse is 10 μ s. During this time the ADC working at 40 MHz ($T = 25$ ns) can collect a sufficient number of samples and perform a reliable comparison vs. V_{TH} . A pulse read of 1 μ s, for example, would not have been long enough to obtain a satisfactory voltage comparison. In Fig. 9 each experimental data point (blue asterisks) is the total energy required to execute a specific SIMPLY core operation (FALSE, SIMPLY when detecting the '00' case and SIMPLY in the other cases) obtained in different experiments at different PW. As expected, the extrapolated trends follow a quasi-linear relation, consisting in a power-law trend with a slope close to one. Although at very short PW values a further reduction of PW may require higher $V_{SET/RESET}$ to maintain a correct circuit behavior, this can be assumed as a solid estimation since these voltage deviations will not follow the exponential trend of PW but will reasonably remain close to 0.7 V or exhibiting a modest increase due to the exponential relation between the device switching speed and applied voltage. The magenta star indicates the projection at 5 GHz, the maximum speed achievable in modern processors, indicating the upper-bound performances that this technology can obtain. For this projection, t_{READ} is equal to 0.1 ns (1 ns/10, with 1ns being the set/reset PW), resulting in a read pulse period of 0.2 ns (5 GHz). We have to point out that, although the read operation is the most frequently performed (and it is executed in 0.2 ns), the SIMPLY architecture will also execute set and reset operations that require 2ns each, although they are executed much less frequently. Therefore, 5 GHz is an indication of the speed of the most frequently performed elementary operation. A more relaxed projection (500 MHz, black circle) is also provided. The same considerations are also adopted for this projection. Results are summarized in Tab. I.

5.2 SIMPLY vs. CMOS

From these promising results, the performances of the LiM implementation of complex Boolean functions can be easily evaluated and compared with the respective CMOS counterparts. In particular, this work presents the performance

comparison between the CMOS 32-bit ripple carry FA, consisting in a cascade of 32 1-bit FA, and the corresponding memristor-based SIMPLY architecture implementation. A 1-bit FA can be realized using SIMPLY, by means of an array of 8 memristors and a sequence of 27 instructions [54]. Despite the required inputs and outputs are only five, at least two additional devices [48] are needed to successfully perform the operation. The considered architecture allows then to derive the energy consumption of the LiM 32-bit FA as the sum of the contributes of each of the 32 1-bit FAs, since each block is directly connected, forming a chain where the carry out (C_{OUT}) of one FA is the C_{IN} of the following. Presented results are comprehensive of the additional energy consumption of a comparator realized in CMOS technology that would be needed in the integrated circuit implementation of the solution. The employed platform does not allow to scale down the comparator energy consumption, because it is fixed by the ADC, but plausible energy consumption values have been already derived from our research group for a low-power integrated comparator [67], and are used here. Tab. II summarizes the full comparison between a memristor LiM SIMPLY based 32-bits FA and its CMOS counterpart, considering also the ideal (and non-implementable) CMOS FA with no energy needed to transmit data from the memory to processing unit and backward [68]–[70]. Also, the comparison between the proposed approach and other experimental [71] and simulated [54], [56], [72], [73] results of LiM-based 32-bit FAs implementations from the literature, further highlights its advantages that are already noticeable at moderate clock frequencies (e.g., 500 MHz). The VNB bottleneck removal by means of this LiM strategy results in a significant energy and delay saving with respect to the traditional CMOS implementation. However, the data reported for the 32-bits CMOS FA without considering the VNB underline that the SIMPLY performance is still to be improved before trying to directly compete with its CMOS counterpart. This is evident in the comparison of the Energy Delay Product (EDP) and Power Delay Product (PDP), important parameters useful to compare and analyze the trade-off between energy saving and performance. The reason behind this can be seen in the many decades of development and research behind the CMOS technology, that led to a continuous improvement of these devices and circuits. Memristor based technology is much younger than CMOS, but its recent development is absolutely promising and will likely lead to a technological revolution in the next decade [6], [74]. However, since the VNB still affects CMOS circuits, the SIMPLY architecture provides an incredible step further in terms of delay and energy savings, with an EDP improvement of seven orders of magnitude in the most reasonably

achievable projection (500 MHz). Even considering the CMOS technology by itself, without the memory wall, the performance is comparable with SIMPLY, especially in terms of energy consumption, the most important parameter for the development of ultra-low power architectures. Despite the visible margins of improvement, the LiM approach, in particular if implemented with SIMPLY, opens an attainable and already accessible way to design a new generation of fast and ultra-low power technology.

CONCLUSIONS

In this work, the possibility of using commercially available and packaged memristor devices has been explored for analog brain inspired and digital LiM computing applications. The observed replicable and stable binary switching, opposed to the found unreliable analog behavior, pushed to the investigation of memristor-based digital architectures which totally remove the necessity of moving data between the memory and the processing unit (VNB). Experimental demonstrations presented in this work suggest that commercial-grade memristor technology can be fully exploited to design digital LiM architectures, based on IMPLY and FALSE operations, allowing a first but important step toward a new generation of ultra-low power computing architectures. To demonstrate the advantages of this technique with respect to CMOS technology, a FPGA-based platform has been designed, built, and tested with commercial-grade packaged SDC-memristors. The study on these devices and their limitations also underlined the advantages of the presented SIMPLY architecture with respect to the traditionally implemented IMPLY. The projections of the energy consumption of a 32-bits FA, obtained by fully demonstrating the truth table of the NAND logic gate at different time conditions, at typical running frequencies of modern integrated logic circuits, proved the significant energy efficiency of this technology compared to the one based on the von Neumann architecture.

Furthermore, the developed platform allows implementing and testing analog and binary NNs, that can coexist with LiM. This provides a prototypical but heterogeneous and flexible architecture supporting the hardware requirements for the envisioned long-term goal of ubiquitous and pervasive artificial intelligence. However, despite this exciting result, the *intrinsic* performance of CMOS technology (i.e., without considering VNB) is still superior, thanks to its continuous improvement during the last 60 years. Memristors development has surely significant margins of improvement, which promises to drive the semiconductor industry toward a new generation of ultra-low power and sustainable computing technology. The demonstration of a working memristor SIMPLY-based platform can open the way for the exploration and implementation of ever more performing systems.

ACKNOWLEDGEMENTS

T. Zanotti and F. M. Puglisi acknowledge financial support from PNRR MUR project ECS_00000033_ECOSISTER.

REFERENCES

- [1] G. E. Moore, "Cramming more components onto integrated circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, pp.114 ff.," *IEEE Solid-State Circuits Society Newsletter*, vol. 11, no. 3, pp. 33–35, Sep. 2006, doi: 10.1109/NSSC.2006.4785860.
- [2] M. T. Bohr and I. A. Young, "CMOS Scaling Trends and beyond," *IEEE Micro*, vol. 37, no. 6, 2017, doi: 10.1109/MM.2017.4241347.
- [3] M. M. Waldrop, "The chips are down for Moore's law," *Nature*, vol. 530, no. 7589, 2016, doi: 10.1038/530144a.
- [4] Wm. A. Wulf and S. A. McKee, "Hitting the memory wall," *ACM SIGARCH Computer Architecture News*, vol. 23, no. 1, 1995, doi: 10.1145/216585.216588.
- [5] J. Backus, "Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs," *Commun ACM*, vol. 21, no. 8, 1978, doi: 10.1145/359576.359579.
- [6] M. Lanza *et al.*, "Memristive technologies for data storage, computation, encryption, and radio-frequency communication," *Science*, vol. 376, no. 6597. American Association for the Advancement of Science, Jun. 03, 2022. doi: 10.1126/science.abj9979.
- [7] S. Oh, H. Hwang, and I. K. Yoo, "Ferroelectric materials for neuromorphic computing," *APL Mater*, vol. 7, no. 9, 2019, doi: 10.1063/1.5108562.
- [8] M. le Gallo and A. Sebastian, "An overview of phase-change memory device physics," *Journal of Physics D: Applied Physics*, vol. 53, no. 21. 2020. doi: 10.1088/1361-6463/ab7794.
- [9] G. Indiveri *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers in Neuroscience*, no. MAY. 2011. doi: 10.3389/fnins.2011.00073.
- [10] E. T. Breyer *et al.*, "Compact FeFET Circuit Building Blocks for Fast and Efficient Nonvolatile Logic-in-Memory," *IEEE Journal of the Electron Devices Society*, vol. 8, 2020, doi: 10.1109/JEDS.2020.2987084.
- [11] G. Indiveri and S. C. Liu, "Memory and Information Processing in Neuromorphic Systems," *Proceedings of the IEEE*, vol. 103, no. 8. 2015. doi: 10.1109/JPROC.2015.2444094.
- [12] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Lett*, vol. 10, no. 4, 2010, doi: 10.1021/nl904092h.
- [13] L. Shi, "Brain inspired computing devices, chips and system," in *2018 Asia-Pacific Magnetic Recording Conference, APMRC 2018*, 2019. doi: 10.1109/APMRC.2018.8601053.
- [14] N. K. Upadhyay, H. Jiang, Z. Wang, S. Asapu, Q. Xia, and J. Joshua Yang, "Emerging Memory Devices for Neuromorphic Computing," *Advanced Materials Technologies*, vol. 4, no. 4. 2019. doi: 10.1002/admt.201800589.
- [15] D. Gandolfi, F. M. Puglisi, G. M. Boiani, G. Pagnoni, K. J. Friston, and J. Mapelli, "Emergence of associative learning in a neuromorphic inference network," 2022, doi: 10.1088/1741-2552/ac6ca7.
- [16] M. A. Zidan, J. P. Strachan, and W. D. Lu, "The future of electronics based on memristive systems," *Nat Electron*, vol. 1, no. 1, 2018, doi: 10.1038/s41928-017-0006-8.
- [17] W. Wan *et al.*, "A compute-in-memory chip based on resistive random-access memory," *504 | Nature |*, vol. 608, 2022, doi: 10.1038/s41586-022-04992-8.
- [18] R. Yang, "In-memory computing with ferroelectrics," *Nature Electronics*, vol. 3, no. 5. 2020. doi: 10.1038/s41928-020-0411-2.
- [19] H. S. P. Wong and S. Salahuddin, "Memory leads the way to better computing," *Nature Nanotechnology*, vol. 10, no. 3. 2015. doi: 10.1038/nnano.2015.29.
- [20] L. O. Chua, "Memristor—The Missing Circuit Element," *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, 1971, doi: 10.1109/TCT.1971.1083337.
- [21] H. S. P. Wong *et al.*, "Metal-oxide RRAM," in *Proceedings of the IEEE*, 2012. doi: 10.1109/JPROC.2012.2190369.
- [22] F. Pan, S. Gao, C. Chen, C. Song, and F. Zeng, "Recent progress in resistive random access memories: Materials, switching mechanisms, and performance," *Materials Science and Engineering R: Reports*, vol. 83, no. 1. 2014. doi: 10.1016/j.mser.2014.06.002.
- [23] T. Zanotti, F. M. Puglisi, and P. Pavan, "Circuit Reliability Analysis of In-Memory Inference in Binarized Neural Networks," 2021. doi: 10.1109/iirw49815.2020.9312858.
- [24] H. Liu, L. Ma, Z. Wang, Y. Liu, and F. E. Alsaadi, "An overview of stability analysis and state estimation for memristive neural networks," *Neurocomputing*, vol. 391, 2020, doi: 10.1016/j.neucom.2020.01.066.
- [25] J. Sun, Y. Wang, P. Liu, and S. Wen, "Memristor-Based Circuit Design of PAD Emotional Space and Its Application in Mood Congruity," *IEEE Internet Things J*, 2023, doi: 10.1109/JIOT.2023.3267778.

- [26] G. Dou, K. Zhao, M. E. I. Guo, and J. U. N. Mou, "Memristor-Based LSTM Network For Text Classification," *Fractals*, 2023, doi: 10.1142/S0218348X23400406.
- [27] J. Sun, Y. Wang, P. Liu, S. Wen, and Y. Wang, "Memristor-Based Neural Network Circuit With Multimode Generalization and Differentiation on Pavlov Associative Memory," *IEEE Trans Cybern*, 2022, doi: 10.1109/TCYB.2022.3200751.
- [28] S. Shirinzadeh, M. Soeken, P. E. Gaillardon, and R. Drechsler, "Logic Synthesis for RRAM-Based In-Memory Computing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 7, 2018, doi: 10.1109/TCAD.2017.2750064.
- [29] S. Kvatinsky *et al.*, "MAGIC - Memristor-aided logic," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, 2014, doi: 10.1109/TCSII.2014.2357292.
- [30] J. Yu, H. A. du Nguyen, M. A. Lebdeh, M. Taouil, and S. Hamdioui, "Enhanced Scouting Logic: A Robust Memristive Logic Design Scheme," in *NANOARCH 2019 - 15th IEEE/ACM International Symposium on Nanoscale Architectures, Proceedings*, 2019, doi: 10.1109/NANOARCH47378.2019.181296.
- [31] P. la Torraca, F. M. Puglisi, A. Padovani, and L. Larcher, "Multiscale modeling for application-oriented optimization of resistive random-access memory," *Materials*, vol. 12, no. 21, 2019, doi: 10.3390/ma12213461.
- [32] F. M. Puglisi, T. Zanotti, and P. Pavan, "SIMPLY: Design of a RRAM-Based Smart Logic-in-Memory Architecture using RRAM Compact Model," in *European Solid-State Device Research Conference*, 2019, doi: 10.1109/ESSDERC.2019.8901731.
- [33] B. Hoffer, N. Wainstein, C. M. Neumann, E. Pop, E. Yalon, and S. Kvatinsky, "Stateful Logic Using Phase Change Memory," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 8, no. 2, 2022, doi: 10.1109/JXCDC.2022.3219731.
- [34] "https://knowm.com/"
- [35] K. A. Campbell, "Self-directed channel memristor for high temperature operation," *Microelectronics J*, vol. 59, 2017, doi: 10.1016/j.mejo.2016.11.006.
- [36] K. A. Campbell, "The self-directed channel memristor: Operational dependence on the metal-chalcogenide layer," in *Handbook of Memristor Networks*, 2019, doi: 10.1007/978-3-319-76375-0_29.
- [37] "Self-Directed channel memristor," <https://knowm.org/memristors/>.
- [38] A. H. Edwards, K. A. Campbell, and A. C. Pineda, "Electron self-trapping in Ge₂Se₃ and its role in Ag and Sn incorporation," in *Materials Research Society Symposium Proceedings*, 2012, doi: 10.1557/opl.2012.1437.
- [39] O. Krestinskaya, K. N. Salama, and A. P. James, "Learning in memristive neural network architectures using analog backpropagation circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 2, 2019, doi: 10.1109/TCSI.2018.2866510.
- [40] G. W. Burr *et al.*, "Neuromorphic computing using non-volatile memory," *Advances in Physics: X*, vol. 2, no. 1, 2017, doi: 10.1080/23746149.2016.1259585.
- [41] G. S. Snider, "Spike-timing-dependent learning in memristive nanodevices," in *2008 IEEE/ACM International Symposium on Nanoscale Architectures, NANOARCH 2008*, 2008, doi: 10.1109/NANOARCH.2008.4585796.
- [42] "https://nebula.wsimg.com/6dba75009009af7a59036365876b3f66?AccessKeyId=64577CB1C10F8DCEF8A3&disposition=0&alloworigin=1."
- [43] J. Taylor and A. Nannarelli, "Design and Simulation of a Quaternary Memory Cell based on a Physical Memristor," in *2016 IEEE Nordic Circuits and Systems Conference (NORCAS), Copenhagen, Denmark, IEEE*, 2016, pp. 1–6.
- [44] I. Marković, M. Potrebić, and D. Tošić, "Memristors as candidates for replacing digital potentiometers in electric circuits," *Electronics (Switzerland)*, vol. 10, no. 2, pp. 1–18, Jan. 2021, doi: 10.3390/electronics10020181.
- [45] B. Linsky and A. D. Irvin, "Principia Mathematica," in *The Stanford Encyclopedia of Philosophy*, Metaphysics Research Lab, Ed., Stanford University, 2020.
- [46] E. Lehtonen and M. Laiho, "Stateful implication logic with memristors," in *2009 IEEE/ACM International Symposium on Nanoscale Architectures, NANOARCH 2009*, 2009, doi: 10.1109/NANOARCH.2009.5226356.
- [47] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, "Memristive switches enable stateful logic operations via material implication," *Nature*, vol. 464, no. 7290, 2010, doi: 10.1038/nature08940.
- [48] E. Lehtonen, J. H. Poikonen, and M. Laiho, "Two memristors suffice to compute all Boolean functions," *Electron Lett*, vol. 46, no. 3, 2010, doi: 10.1049/el.2010.3407.
- [49] S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Memristor-based material implication (IMPLY) logic: Design principles and methodologies," *IEEE Trans Very Large Scale Integr VLSI Syst*, vol. 22, no. 10, 2014, doi: 10.1109/TVLSI.2013.2282132.
- [50] F. M. Puglisi, L. Pacchioni, N. Zagni, and P. Pavan, "Energy-efficient logic-in-memory I-bit full adder enabled by a physics-based RRAM compact model," in *European Solid-State Device Research Conference*, 2018, doi: 10.1109/ESSDERC.2018.8486886.
- [51] A. Raghuvanshi and M. Perkowski, "Logic synthesis and a generalized notation for memristor-realized material implication gates," in *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*, 2015, doi: 10.1109/ICCAD.2014.7001393.
- [52] T. Zanotti, F. M. Puglisi, and P. Pavan, "Circuit reliability of low-power rram-based logic-in-memory architectures," in *IEEE International Integrated Reliability Workshop Final Report*, 2019, doi: 10.1109/IIRW47491.2019.8989875.
- [53] F. M. Puglisi, T. Zanotti, and P. Pavan, "Unimore resistive random access memory (rram) verilog-a model," <https://nanohub.org/publications/289/1>, 2019.
- [54] T. Zanotti, F. M. Puglisi, and P. Pavan, "Smart Logic-in-Memory Architecture for Low-Power Non-Von Neumann Computing," *IEEE Journal of the Electron Devices Society*, vol. 8, 2020, doi: 10.1109/JEDS.2020.2987402.
- [55] C. Nguyen *et al.*, "Advanced 1T1R test vehicle for RRAM nanosecond-range switching-time resolution and reliability assessment," in *IEEE International Integrated Reliability Workshop Final Report*, 2016, doi: 10.1109/IIRW.2015.7437059.
- [56] T. Zanotti, P. Pavan, and F. M. Puglisi, "Multi-input logic-in-memory for ultra-low power non-von neumann computing," *Micromachines (Basel)*, vol. 12, no. 10, 2021, doi: 10.3390/mi12101243.
- [57] Z. Zhao *et al.*, "A Memristor-Based Spiking Neural Network with High Scalability and Learning Efficiency," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 5, 2020, doi: 10.1109/TCSII.2020.2980054.
- [58] T. Kim *et al.*, "Spiking Neural Network (SNN) With Memristor Synapses Having Non-linear Weight Update," *Front Comput Neurosci*, vol. 15, 2021, doi: 10.3389/fncom.2021.646125.
- [59] L. A. Camuñas-Mesa, B. Linares-Barranco, and T. Serrano-Gotarredona, "Neuromorphic spiking neural networks and their memristor-CMOS hardware implementations," *Materials*, vol. 12, no. 7, 2019, doi: 10.3390/ma12172745.
- [60] Y. Kim *et al.*, "Memristor crossbar array for binarized neural networks," *AIP Adv*, vol. 9, no. 4, 2019, doi: 10.1063/1.5092177.
- [61] T. van Nguyen, J. An, and K. S. Min, "Memristor- CMOS hybrid neuron circuit with nonideal-effect correction related to parasitic resistance for binary-memristor-crossbar neural networks," *Micromachines (Basel)*, vol. 12, no. 7, 2021, doi: 10.3390/mi12070791.
- [62] Z. Zhang, Z. Ge, Y. Wei, X. Cheng, G. Xie, and G. Liu, "1S-1R array: Pure-memristor circuit for binary neural networks," *Microelectron Eng*, vol. 254, 2022, doi: 10.1016/j.mee.2021.111697.
- [63] T. Zanotti, P. Pavan, and F. M. Puglisi, "Study of RRAM-Based Binarized Neural Networks Inference Accelerators Using an RRAM Physics-Based Compact Model," in *Neuromorphic Computing*, 2023, [Online]. Available: www.intechopen.com
- [64] T. Zanotti, F. M. Puglisi, and P. Pavan, "Reliability and Performance Analysis of Logic-in-Memory Based Binarized Neural Networks," *IEEE Transactions on Device and Materials Reliability*, vol. 21, no. 2, 2021, doi: 10.1109/TDMR.2021.3075200.
- [65] L. Xie *et al.*, "Scouting Logic: A Novel Memristor-Based Logic Design for Resistive Computing," in *Proceedings of IEEE Computer Society Annual Symposium on VLSI, ISVLSI*, 2017, doi: 10.1109/ISVLSI.2017.39.

- [66] S. Kvatinsky *et al.*, "MAGIC - Memristor-aided logic," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, 2014, doi: 10.1109/TCSII.2014.2357292.
- [67] T. Zanotti *et al.*, "Reliability of Logic-in-Memory Circuits in Resistive Memory Arrays," *IEEE Trans Electron Devices*, vol. 67, no. 11, 2020, doi: 10.1109/TED.2020.3025271.
- [68] T. Zanotti, F. M. Puglisi, and P. Pavan, "Reliability-Aware Design Strategies for Stateful Logic-in-Memory Architectures," *IEEE Transactions on Device and Materials Reliability*, vol. 20, no. 2, 2020, doi: 10.1109/TDMR.2020.2981205.
- [69] S. Y. Park, D. Jung, J. U. Kang, J. S. Kim, and J. Lee, "CFLRU: A replacement algorithm for flash memory," in *CASES 2006: International Conference on Compilers, Architecture and Synthesis for Embedded Systems*, 2006. doi: 10.1145/1176760.1176789.
- [70] M. Aguirre-Hernandez and M. Linares-Aranda, "CMOS full-adders for energy-efficient arithmetic applications," *IEEE Trans Very Large Scale Integr VLSI Syst*, vol. 19, no. 4, 2011, doi: 10.1109/TVLSI.2009.2038166.
- [71] L. Cheng *et al.*, "Reprogrammable logic in memristive crossbar for in-memory computing," *J Phys D Appl Phys*, vol. 50, no. 50, 2017, doi: 10.1088/1361-6463/aa9646.
- [72] N. Talati, S. Gupta, P. Mane, and S. Kvatinsky, "Logic design within memristive memories using memristor-aided loGIC (MAGIC)," *IEEE Trans Nanotechnol*, vol. 15, no. 4, 2016, doi: 10.1109/TNANO.2016.2570248.
- [73] A. Siemon *et al.*, "Stateful Three-Input Logic with Memristive Switches," *Sci Rep*, vol. 9, no. 1, 2019, doi: 10.1038/s41598-019-51039-6.
- [74] D. V. Christensen *et al.*, "2022 roadmap on neuromorphic computing and engineering," *Neuromorphic Computing and Engineering*, vol. 2, no. 2, 2022, doi: 10.1088/2634-4386/ac4a83.