

Practical Implementation of Memristor-Based Threshold Logic Gates

G. Papandroulidakis¹, A. Serb¹, *Member, IEEE*, A. Khiat¹, G. V. Merrett², *Senior Member, IEEE* and T. Prodromakis¹, *Senior Member, IEEE*

¹Electronic Materials & Devices Research Group, Zepler Institute for Photonics and Nanoelectronics, University of Southampton, SO17 1BJ, UK

²Cyber Physical Systems Research Group, School of Electronics and Computer Science, University of Southampton, SO17 1BJ, UK
g.papandroulidakis@soton.ac.uk

Abstract—Current advances in emerging memory technologies enable novel and unconventional computing architectures for high-performance and low-power electronic systems, capable of carrying out massively parallel operations at the edge. One emerging technology, ReRAM, also known to belong in the family of memristors (memory resistors), is gathering attention due to its attractive features for logic and in-memory computing; benefits which follow from its technological attributes, such as nanoscale dimensions, low power operation, and multi-state programming. At the same time, design with CMOS is quickly reaching its physical and functional limitations, and further research towards novel logic families, such as Threshold Logic Gates (TLGs) is scoped. In this paper, we introduce a physical implementation of a memristor-based current-mode TLG (MCMTLG) circuit and validate its design and operation through multiple experimental setups. We demonstrate 2-input, 3-input, and 4-input MCMTLG configurations and showcase their reconfiguration capability. This is achieved by varying memristive weights arbitrarily for shaping the classification decision boundary, thus showing promise as an alternative hardware-friendly implementation of Artificial Neural Networks (ANNs). Through the employment of real memristor devices as the equivalent of synaptic weights in TLGs, we are realizing components that can be used towards an in-silico classifier.

Keywords—*memristor, Artificial Neural Networks, Threshold Logic Gates, in-silico Classifiers, Reconfigurable electronics*

I. INTRODUCTION

Today's conventional computing paradigm is based on the MOSFET transistor and CMOS technology; two cornerstones which have underpinned the development of digital electronics over the last 5 decades. Although there is still optimism for future improvement of CMOS, accumulating scientific evidence indicates the need for advances in both new emerging technologies to replace MOSFETs and in new computer circuits

and architectures [1], [2]. The former addresses the increasing difficulty of pursuing further downscaling (with its associated drop in reliability [2]) whilst the latter seeks to address the Von Neumann bottleneck, where increasingly large memories and powerful processors struggle to communicate over a limited interlink whose data transfer capacity doesn't scale fast enough [2]–[4].

On the technology front, recent advances in emerging memory technologies introduce new tools in electronic system design. One prominent technology, ReRAM devices [5] (part of the memory resistor, memristor, family of devices) can act as nanoscale [6], finely tuneable [7], electrically programmable [8], [9] resistive elements. Memristors are capable of storing multi-bit information and retaining their memory state when powered off (non-volatile) while simultaneously their adoption in electronics is accelerated by additional advantages they provide, such as better area scaling, low power consumption and CMOS-compatibility [10]–[12]. Hence, memristor devices are considered one of the most promising candidates for the next generation of computer circuits, systems and architectures [13]–[16]. This enables the development of area and power efficient reconfigurable electronics, which are very important in a wide range of applications, e.g. the embedded computing systems that process the data at the edge, where there is a continuous race towards minimization of chip area and power consumption for neuromorphic edge computing [17], [18].

On the computation/architecture front, there has been a sustained effort to develop bio-inspired computation concepts, mostly in the guise of artificial neural network (ANN)-enabled systems. Research on artificial neural networks has thus far spanned the entire interval between the first simplified models of all-or-none hardware neurons [19] and the current state-of-the-art GPU-based ANNs [20]–[22]. However, one often overlooked example of ANN-like computation can be found in

Manuscript was submitted for review on 03 October 2018. This research is supported by the Engineering and Physical Sciences Research Council EP/K017829/1 research grant. All supporting data are available through the University of Southampton repository with dataset DOI: <https://doi.org/10.5258/SOTON/D0824>

Georgios Papandroulidakis is a Ph.D candidate with Zepler Institute for Photonics and Nanoelectronics, University of Southampton, SO17 1BJ, UK (email: g.papandroulidakis@soton.ac.uk).

Alexander Serb is a Research Fellow with Zepler Institute for Photonics and Nanoelectronics, University of Southampton, SO17 1BJ, UK (email: a.serb@soton.ac.uk).

Ali Khiat is an Experimental Officer with Zepler Institute for Photonics and Nanoelectronics, University of Southampton, SO17 1BJ, UK (email: a.khiat@soton.ac.uk).

Geoff V. Merrett is an Associate Professor with School of Electronics and Computer Science, University of Southampton, SO17 1BJ, UK (email: gvm@ecs.soton.ac.uk).

Themis Prodromakis is a Professor with Zepler Institute for Photonics and Nanoelectronics, University of Southampton, SO17 1BJ, UK (email: t.prodromakis@soton.ac.uk).

the form of its quantized, digital counterpart, the so-called threshold logic (TL) [19]. TL is a model for performing a comparison between a threshold value and the weighted sum of an input vector. A basic computational unit in TL is called a threshold logic gate (TLG) and it corresponds to the artificial neuron in ANNs. TLGs were introduced as a method of describing and modeling neural activity in the brain through conventional electronic circuits and systems [19] [23]. Although TL is effectively a simplification of ANNs, TLG-based logic families have been shown to be capable of fast and low-power operation as evaluated by the power-delay trade-off metric [24]–[26]. Many of these implementations suggest the hardware design of ANN-like circuits and systems could greatly benefit from using TLGs as their fundamental logic cell, thus enabling advances towards neuromorphic computer architectures and applications, as some recent work explores [27]–[30].

In this work, we demonstrate a practical implementation of the metal-oxide memristor-based differential TLG (MCMTLG) circuits, thus laying the foundation for building artificial perceptron networks. In Section II, we provide the background that supports the concept of memristive TLGs as the basis for future reconfigurable computing systems. In Section III, we provide the design and operation description of the MCMTLG. In Section IV, we explain the experimental setup, and in Section V, we validate the functionality of the proposed gates through 2-input, 3-input and 4-input experimental setups where tuneable memristor weights are used as artificial synaptic weights, thus defining the contribution of each input component to the TLG's comparison function. Notably, this functionality is enabled by the recently introduced multi-bit memristor technology [7], which enables fine, continuous control of the memristive synaptic weights. We show experimentally how changes in memristor resistances affect the shape of the decision boundary of the MCMTLG and comment on key, practical factors that affect performance. In Section V, we modeled the proposed MCMTLG circuit into Cadence's Virtuoso Spectre simulation environment and performed performance testing to provide estimation regarding its power and delay metric. Through the use of the modeled MCMTLG, we performed a comparison with existing memristive TL circuits (Section V). Finally, our conclusions and discussion of future work is presented in Section VI.

II. TLG AND MEMRISTOR TECHNOLOGY BACKGROUND

Many competing emerging memory technologies are part of the memristor technology family, such as PCM, ECM, VCM etc. [31], [32]. A number of these technologies have been studied as novel reconfigurable circuit and system components [13], [33]–[35] (including TLGs [30], [36]–[38] – exclusively simulated). In principle, any memristor technology featuring non-volatile resistive switching, sufficiently high ON/OFF ratio and not excessively high or low resistance levels can be introduced into an appropriately designed TLG. In this work, we have used a type of metal-oxide-based memristive devices shown to routinely support analogue, non-volatile resistive state resolution of 5 bits [7].

Different designs of TLGs have been proposed with different trade-offs regarding their power-noise ratio performance [26] and different circuit implementations [26], [39], [40]. Recently,

the use of differential type Current-Mode (CM) TLGs (CM-TLGs) seems to gain ground as one of the fastest and low-power TL implementations [26], [41]. More importantly, recent work is showcasing optimized differential CM-TLGs [42]–[44], highlights the benefits this logic technology offers in comparison with conventional CMOS logic networks regarding area and power savings [25].

The transistor-based differential CM-TLGs can provide some competitive solutions, but at the same time, their benefits are limited by increased sensitivity to noise and device mismatch, as well as limited fan-in (input vector space dimensionality) [45], [46]. Transistor-based TLG designs are regulating the input weights and classifying the input space through parallel connections of MOS devices. These designs were limited by the need to use a standard minimum transistor size as the unity input contribution multiplier [42]. For applying different non-unity multiplier factors per input, multiple transistor components needed to be selected. At the same time, the transistor-based TLGs are incapable of representing non-integer contribution values, as the weights are determined by a specific number of binary weights [44], [47].

Towards providing solution to these limitations a lot of work has focused into novel and more efficient implementations of the differential synaptic weight circuits, by introducing conventional electronics, such as capacitors and resistors [40], [48], while other recent implementations take advantage of emerging nano-electronics devices such as single-electron technology (SET) [49] and negative resistance devices (NRD) [50], [51]. More recently, Rajendran et al. [52] and later Dara et al. [41], among others, have shown that memristor technology can efficiently be incorporated into TLG designs, hence becoming the catalyst of significant power consumption and noise sensitivity reduction, as well as logic and area scaling in TLGs, compared to conventional Boolean logic gate. Introducing the memristor devices as analogue weights in a digital logic gate family, has the advantage of enabling highly localized, continuously tuneable, minimal front-end footprint and low-voltage operated non-volatile memory into the TLG, thus providing a potentially decisive advantage in the implementation of memory-heavy ANN accelerators [30], [38], [53]–[56].

From the available memristive TL implementations, the computing schemes of differential memristively-enhanced load comparison TLGs are shown to have advantages over simpler memristor-based TLG designs [53], [57], [58]. More specifically, the differential implementations, in general [39], showcase delay and energy improvements over non-differential memristive TL (MTL) designs [59]–[62]. At the same time, there is a significant trade-off of energy-delay-flexibility and area-complexity between these two main groups of TLGs. While the differential TLG group is optimized for performance and logic-centric features (e.g. positive and negative weight configurations as in [43]) the non-differential MTL gates provide a simpler gate structure where a resistive network (weighted inputs) is fed into an inverter (thresholding element). Different variants of the MTL scheme are available and can provide novel solutions to memristively-enhanced TLG-based computer architectures, capable of competing against conventional systems in applications such as object recognition, FPGAs, etc. [28]–[30], [60], [63]–[65].

Memristively-enhanced TLGs have a lot of competition from other memristor-based logic circuits (depending on the application). Technologies such as memristor-based Look-Up-Tables (LUTs) [66]–[68] and memristor-based universal logic gates [69]–[71], such as memristor ratioed logic (MRL) NAND gates [71], may be preferable in some architectures and/or applications over TLGs. But the MCMTLG’s requirements for a state-of-the-art memristor technology, such as the one used in this work, favours in critical ways the implementation of non-uniformly behaved programmable analogue resistive elements. At the same time, TL computing schemes do not require frequent switching of the memristive weights, as they are mainly used as programmed-once-read-many types of reconfigurable logic, thus do not impose high requirements of switching endurance in memristors. In contrast, LUTs techniques require stable and hard-defined memristive states to operate correctly, while other logic techniques that make use of memristive networks to perform state-based logic [71] requires total homogeneity of device behaviour and high endurance in large crossbar arrays to be viable as true alternative post-von Neumann solution. While unorthodox by mainstream conventional systems, the implementation of future computers that make use of non-uniformly behaved components might be the key to a new era of computing. Neuro-inspired logic schemes, such as TLGs, are ideal to ‘assimilate’ such ‘imperfect’ technologies, i.e. technologies that do not offer better reliability and controllability compared with existing conventional digital electronics, and use them to build new generations of computers, similar to what biological brains seems to achieve in nature’s biological neural networks.

Combining an understanding of TLGs as fundamental computational units in ANN-like post-von Neumann computing schemes with the recently demonstrated multi-bit capabilities and fine tuning of memristances of metal-oxide-based memristive devices raises the prospect of a memristor-based reconfigurable fabric. In the following section, we present our approach into memristively enhanced CM-TLGs (MCMTLG) and initial experimental results using a discrete component-on-breadboard circuit implementation of the proposed design.

III. PROPOSED MCMTLG DESIGN AND OPERATION

A differential CM-TLG design, such as the MCMTLG, consists of two parts (see Fig.1), the differential and the sensor part. The differential part consists of the input and threshold branches, handling the input and threshold memristance input vectors respectively. Within each branch, the weight vectors are implemented by a bank of 1T1M (memristively source-degenerated pMOS transistors) ensembles. Each 1T1M ensemble receives a digital input signal controlling the gate of the pMOS transistor; a single element of the branch’s input vector, with the accompanying memristor defining the contribution of each such vector element. If the input is low (active), then a memristance-dependent current flows from that 1T1M sub-branch towards the sensor part. Additionally, each of the differential branches is power-gated by a back-to-back (BtB) pMOS circuit.

The sensor part is the thresholding element of the circuit, comparing the differential inflowing currents and settling to a binary output indicating which is greater. It is designed as an SRAM memory cell; a latching element consisting of two BtB

connected CMOS inverters, forming a positive feedback loop (Fig.1). Furthermore, two additional CMOS inverters are added at the outputs of the sensor part, one per output, hence avoiding any voltage level degradation and isolating the sensor part from the circuitry connected further down the logic cascade. The power supply to the sensor part with the isolation inverters is controlled by a pMOS power gate.

The main design features of our proposed MCMTLG are based on a variety of differential TL circuits, such as [41], [42]. Similar to Dual Clocked Current Mode Threshold Logic Gate (DCCML) design, presented in [43], we used a common voltage supply for both sensor and differential parts. Furthermore, the differential 1T1M banks were connected to the outputs of the sensor, thus speeding up the sensor’s decision-making operation (differential current comparison) by removing the RC paths introduced by the 1T1M array path, a design feature similar to transistor-based coupled inverters with asymmetrical loads (CIAL) [72], threshold logic CIAL (CIAL-TL) [73] and their memristor-based counterpart from Dara et al. [41].

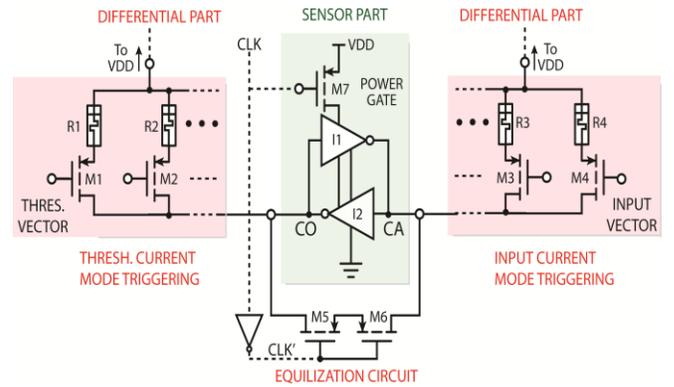


Fig. 1. Memristor-enhanced Threshold Logic Gate (MCMTLG) concept schematic. The two basic parts are the memristor-based 1T1M array performing a dot-product multiplication between the memristor state (memristance, i.e. memory resistance) and the binary input vector controlling the accompanying pMOS device, and the sensor determining which 1T1M array outputs greater current. CA and CO nodes, standing for canonical (CA) and complementary (CO), respectively, are the output nodes of the gate. The outputs are available during the evaluation phase when the differential current flows have been compared and a final stable state of the sensor part has been obtained. The clock signal (CLK) and its complement (CLK’) are controlling the sensor’s power gate and the equalization circuit, respectively. Hence, the CLK signal defines the transition between the two operation phases, equalization (reset) and evaluation (set).

The MCMTLG circuit performs a current comparison operation in two phases. During the equalization phase, the differential part is power-gated on and the sensor part is power-gated off, thus forcing the sensor part into an unstable equilibrium. In that phase, the voltages at CO and CA are forced to be almost equal by the shunting BtB pMOS devices between the branches. Next, in the evaluation phase, the inter-branch shunting is released, the differential part is power-gated off and the sensor part is power-gated on. This allows differences on nodes CO and CA to be amplified by the positive feedback action of the BtB-connected inverters of the sensor part [45]. Notably the differential part is cut-off from the voltage supply during the evaluation phase, thus disabling the current flow through the 1T1M input and threshold arrays and towards the

sensor, thus leaving only a brief window for the sensor (during the short period of CLK falling and CLK' rising) of achieving a stable and correct transition to a binary memory state, based on the small voltage differences settled during the equalization.

IV. EXPERIMENTAL SETUP

For this practical memristor-enhanced circuit implementation we used memristor devices designed and fabricated in-house by our group. All the memristors used in the experimental setups are in $3\times 3\text{mm}^2$ chips that are wire-bonded to PLCC68 packages. Each memristor is a $60\times 60\ \mu\text{m}^2$ cross-point of the top and bottom electrodes. All circuits implemented throughout this work rely on the rich dynamics of an in-house metal-oxide ReRAM technology employing metal-insulator-metal (MIM) devices [7].

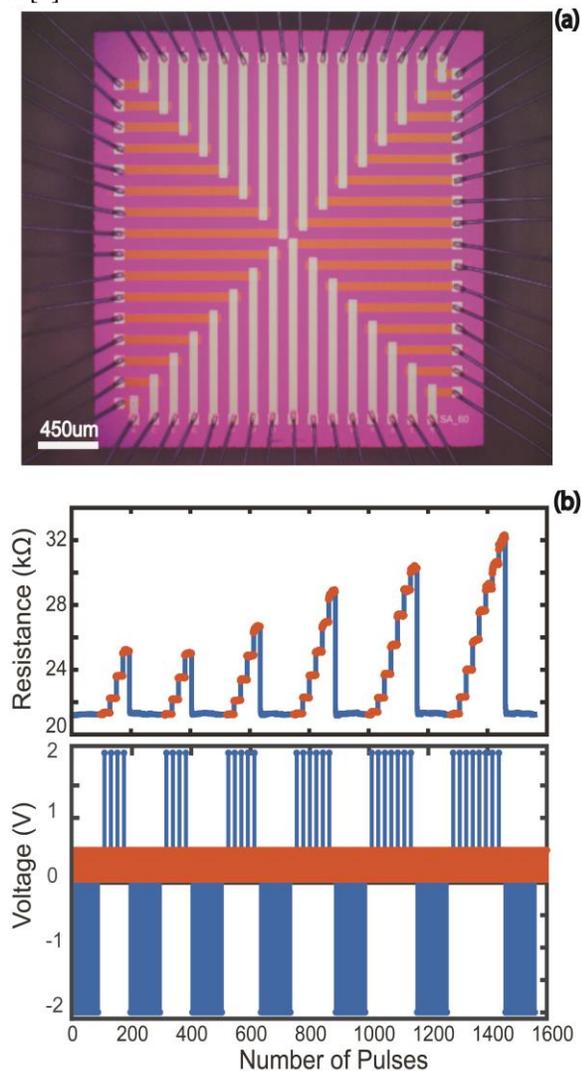


Fig. 2. Characteristic behavior of the memristor devices. In Fig.1a a standalone memristive topology is shown, where each crosspoint device is a memristor. Each device is a stack of Pt/Al₂O₃/TiO₂/Pt/Ti (10/4/24/10/5) nm, with the dimensions per crosspoint device being $60\times 60\ \mu\text{m}^2$. In Fig.1b an example of arbitrary programming of select memristive states is presented [8]. The upper trace of (b) presents the memristance programming due to the voltage bias pulse, shown in the lower trace of (b). The memristors are fabricated in-house by our team. Alongside the +2V bias pulses corresponding to the multi-state memristance programming, we can observe the +0.5V reading pulses, showcased in orange color, as well as the -2V pulses that assure us that before each programming phase the memristance is resetted fully (Fig.2 is adapted from [75]).

Originally, the devices were fabricated on 6-inch SiO₂/Si wafer with the bottom and top electrodes (BEs & TEs) patterned using optical lithography, e-beam evaporation, and lift-off processes. Similar processes were adopted for the active layer patterning, except that sputtering was used for the deposition with a magnetron-sputtering tool. The active layer is constituted of TiO₂ and Al₂O₃ thin-film metal-oxides. After dicing, $3\times 3\text{mm}^2$ wire-bonded chips containing memristor devices were obtained, with MIM stacks constituting of Pt/Al₂O₃/TiO₂/Pt/Ti (10/4/24/10/5) nm. Fig.2a shows an example of a chip that contains 32 stand-alone devices. Scalability is another advantage of memristors when used in a cross-bar array configuration. Pi et al. have demonstrated devices down to $2\times 2\ \text{nm}^2$ with 12 nm pitch [74].

All hardware experiments presented through this work were carried out on circuits prototyped on breadboard. External power supply was used to supply the power rail of the implemented circuits. The results were gathered exclusively by an oscilloscope. For these experiments, packaged devices were used, connected to the breadboard discrete component-based circuit using a breakout board. The power supply used for the experiments was 0.65V, to avoid any unwanted state programming through the trains of reading pulses applied to the differential part of the circuit. For the pMOS devices, we used the NDP5020P (1H10AA) model while for the nMOS devices we used the SUP85N02-03 (T32BAA) model.

The memristor devices used throughout these experiments are detailed above. Prior to use in the practical proof-of-concept MCMTLG case studies, the devices were pre-conditioned separately by electroforming and resistance stabilization in the required functional range. We measured the circuit response through the Rigol MSO4000 Oscilloscope. The input vector and the clock signal were produced through microcontroller programming and converted to circuit-specific voltage levels through custom resistive voltage divider circuits. It worth noting that for the case of 3-input we measured the input vectors and the clock signal through the Logic Analyser (LA) digital probes, due to the limited number of analogue probes available on the oscilloscope. In each experiment, the memristive devices used were programmed in the required state using an ArC ONE instrument board (ArC Instruments, UK). An example of such programming is provided through Fig.2b [7]. All devices used for all the experiments were located on the same die, i.e. only one memristive device package containing a total of 32 memristors. Having decided upon the details of the components of our practical implementation, we demonstrate experimental setups to validate the functionality of the circuit and gaining insights regarding its real-world constrained operation.

V. MCMTLG VALIDATION

The proposed MCMTLG was built using discrete components on a breadboard. For our physical implementation, the differential circuit uses two pMOS transistors back-to-back for power-gating (see Fig. 4a). This is done to control the connection of the input and threshold weighted vectors to the voltage supply, thus avoiding logic state degradation of the latching element during the evaluation phase, as well as improved operational stability even with noisy input vectors. The BtB pMOS circuits also enable lower power consumption, due to the fact that the differential part is cutoff and does not

consume power during the evaluation phase. The voltage supply V_{DD} used in the proposed design is 0.65V, thus ensuring that the memristive devices being used cannot be accidentally programmed during operation. Furthermore, a BtB PMOS circuit was used also for the equalization circuit that reset the sensor part before the evaluation being performed. V_{CLK} , which control the operation cycle of equalization/evaluation, and the input vector's high voltage level, are set to 0.9V. The low logic level for both the V_{CLK} and the input vector is set to 0V (Gnd). A microcontroller (Raspberry Pi 3 Model B) is used to generate the clock signal as well as the input vector used for all the experiments (2-, 3- and 4-input MCMTLGs) described below.

A. 2-input MCMTLG

In the 2-input circuit experiments, the threshold branch consists of a single 1T1M sub-circuit (TH) while the input vector consists of two 1T1M sub-circuits (M1, M2). Simultaneously, AND and OR functionality, can be interpreted as different flavours of majority-gating, thus defining a decision boundary that classifies the input vector (see Fig.3a,b). For example, a MAJ-1 gate is equivalent to a 2-input OR gate and a MAJ-2 is homomorphic to an AND gate. This implies memristive weights set in such a way that either input has a larger weight than the threshold branch (i.e. $M1, M2 < TH$). Similarly for AND: $M1, M2 > TH$, but $M1 || M2 < TH$. Hence, by altering the memristances of the devices we are able to alter the functionality of the MCMTLG. An LTSpice simulation of a TL using resistors of various values as weights demonstrates the effect of resistance on the decision boundary of the TL (Fig. 3(c)).

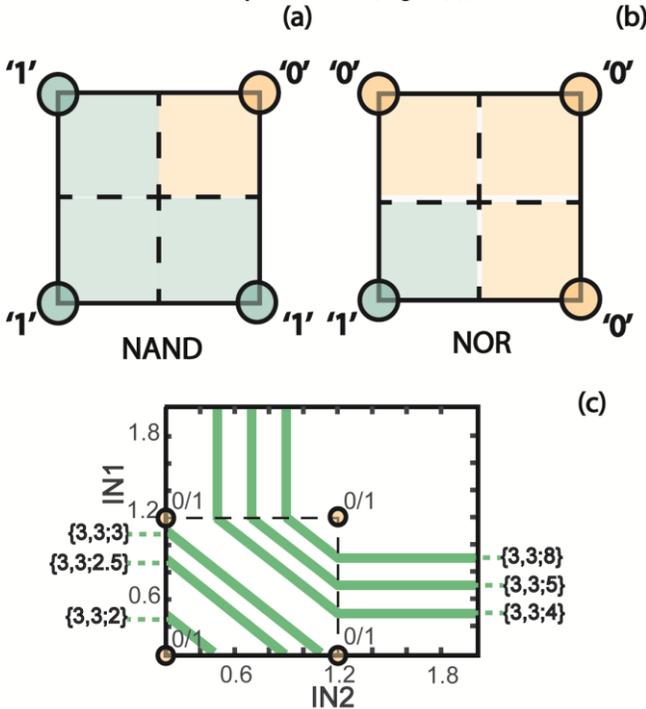


Fig. 3. Concept-level figures highlighting the space separation performed by TLGs and the programmability of the threshold. (a), (b) Indicative 2D input space splitting performed by the 2-input MCMTLG, for the NAND case (a) and the NOR case (b) respectively. (c) LTSpice simulation of an MCMTLG using resistors instead of memristors. Changing the threshold weight alters the decision boundary of the gate. Each weight configurations denoted on the side of each case, in the form of {a, b; c}, in units of $M\Omega$, where a, b and c are the input/threshold resistive weights of each configuration.

In the circuit level, the aforementioned differential impedance enables a majority functionality, where at least one of the input 1T1M sub-circuits has to be conductive in order for the input network conductance to enable higher current flow toward the sensor part, thus enforcing a change in the memory state of the CMOS latching element. At the same time, the value of the input memristive weights are used to define the contribution of each input or threshold, thus used as a form of signal multipliers where a lower memristive weight results in a higher multiplier factor for an input/threshold signal compared to a higher memristive weight. Through the employment of these programmable contribution rates, we are able of enhancing the reconfiguration capabilities of the circuit.

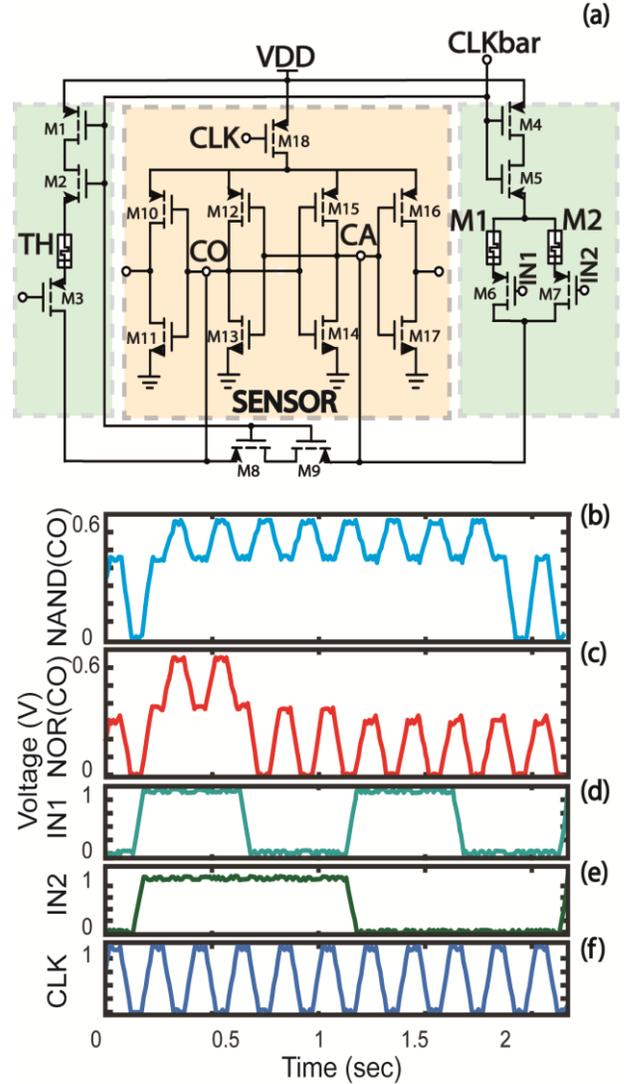


Fig. 4. The circuit schematic and the measured response of the practically implemented 2-input Memristive Current Mode Threshold Logic Gate (MCMTLG). (a) Circuit schematic of the practically implemented Memristive Current Mode Threshold Logic Gate (MCMTLG). (b) Experimentally measured complementary output illustrating NAND functionality (AND/NAND weight configuration). (c) NOR functionality as measured from the CO output (OR/NOR weight config.). (d, e) Input vector signals. (f) Clock signal. Memristor configurations: $\{M1, M2; TH\} = \{60.5k\Omega, 60k\Omega; 33k\Omega\}$ $\{33k\Omega, 18.3k\Omega; 41.7k\Omega\}$ for the AND/NAND and OR/NOR function, respectively. Due to the use of pMOS devices as the transistor part of the 1T1M branches, the HIGH and LOW voltage level, of the input vector, corresponds to non-conductive and conductive transistor device, respectively.

For the physical implementation of a 2-input AND MCMTLG we used the weight configuration of $\{M1, M2; TH\} = \{60.5k\Omega, 60k\Omega; 33k\Omega\}$, thus satisfying the requirements for the AND TL inequality equation (majority-2 function). These values are chosen from the available dynamic range of the memristor devices programmability [7]. From the canonical (CA) output we can measure the AND function circuit response, while from the complementary (CO) output we can obtain the complementary function, NAND. The measured response of the AND/NAND TLG configuration is shown in Fig.4b and the input vector is presented in Fig.4d for the first input (IN1) and in Fig.4e for the second input (IN2). The clock signal that controls operation is shown in Fig.4f. Due to the use of binary input vectors the quantized corner points of the 2D area, as seen in Fig.3a,b, are of interest.

The CA (canonical) output, where AND and OR functions can be measured, and CO (complementary) output, where the NAND and NOR functions can be measured, can be seen in Fig.4a. Fig 4b showcase the complementary (CO) output of the circuit configured to perform the 2-input AND/NAND gate (NAND(CO)), while in Fig.4c the measured CO output for the OR/NOR configuration (NOR(CO)) is shown. The clock signal is determining the equalization/reset phase (clock HIGH) and evaluation/set phase (clock LOW) cycle of operation. The outputs NAND(CO) and NOR(CO) are valid during the evaluation phase, while during the equalization we can see that the output signals stay at an intermediate unstable logic level. The $V_{DD}=0.65V$ and the $V_{CLK}=V_{IN}=0.9V$ (for the logic '1'). It is worth noting that due to the use of pMOS devices in the 1T1M sub-circuits of the differential part, the logic for HIGH input voltage the corresponding input is non-conductive (logic '0') while for LOW input voltage the corresponding input is conductive (logic '1').

In the case of the OR functioning MCMTLG the differential part configuration was set as $\{M1, M2; TH\} = \{33.8k\Omega, 18.3k\Omega; 41.6k\Omega\}$. These values are chosen from the available dynamic range of the memristor programmability. Similarly to the configuration of the AND/NAND case study, the input vector (Fig.4d,e) and the clock signal (Fig.4f) are the same for this operational configuration. Both OR and NOR, as well as the AND/NAND, functions are performed simultaneously due to the complementary bi-stable operation of the sensor part.

Towards getting a detailed glimpse of the inner working of the physically implemented circuit, we are demonstrating a practical experiment to validate a 2-input MCMTLG with sweeping inputs, as shown in Fig. 5. The circuit has been developed for binary input space (the input vector control the accompanying transistors of the 1T1M) and responds with binary values, but we are able to capture the full behavioural aspect of the MCMTLG by introducing analogue ramps as inputs to its input 1T1M array.

As shown in the results of this experiment we can map the evaluation response of a 2-input MCMTLG. For the four cases of the response in the analogue input signal, we used 3 memristor devices per gate configuration, one for the threshold 1T1M array and 2 for the input 1T1M array. We program the memristors for performing all the possible TL function given the input space (2 inputs) under test. The possible function are the AND ($M1, M2 > TH$ and $M1 || M2 < TH$) (see Fig.5a), OR ($M1, M2 < TH$) (see Fig.5b), IN1 ($M1 < TH, M2 > TH$) (see Fig.5c)

and IN2 ($M1 > TH, M2 < TH$) (see Fig.5d). From each case, shown in Fig. 5, highlighting the different responses of the MCMTLG for different memristance weight sets, we are observing the moving of the 2D space separation boundary. For the cases in Fig. 5a,b we can understand the separation boundary as a moving 45° line, while for the case in Fig. 5c,d the separation boundary has the form of 90° and 180° lines, respectively.

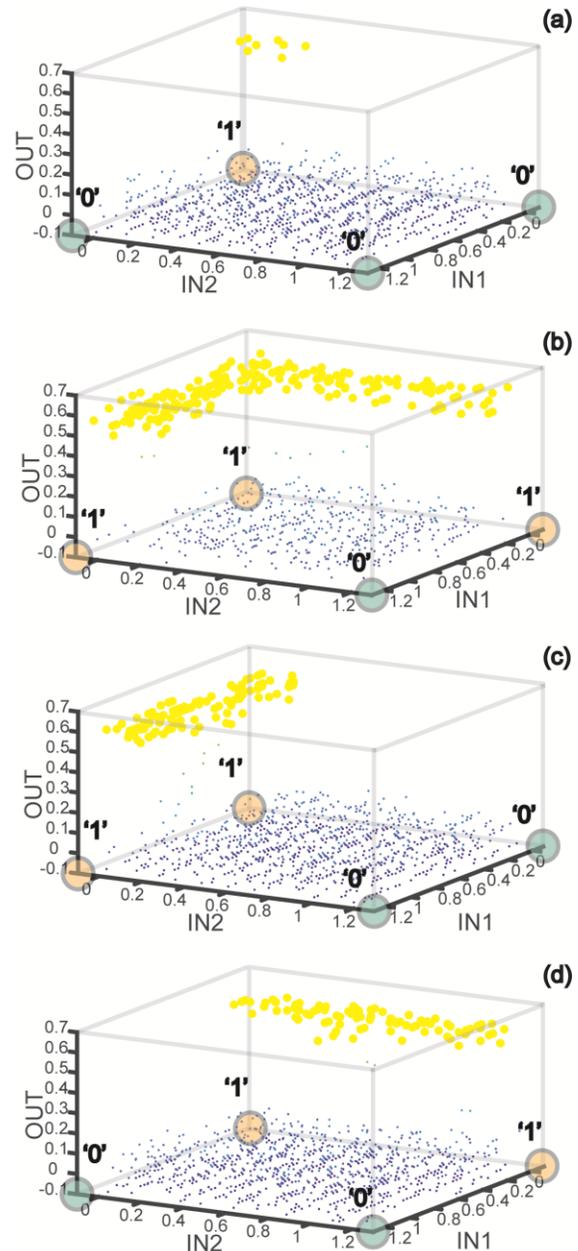


Fig. 5. Experimental validation of the way the MCMTLG evaluates and maps a 2-input vector to the binary output space. By using analogue inputs (ramps from 0V to 1.2V per input) we create a clearer map of the circuit response. Four different cases are presented in this experiment, mapping every possible TL boolean function. (a) AND TLG ($M1, M2 > TH$ and $M1 || M2 < TH$), (b) OR ($M1, M2 < TH$), (c) $OUT=IN1$ ($M1 < TH, M2 > TH$) and (d) $OUT=IN2$ ($M1 > TH, M2 < TH$). The memory configurations for each case of the experiment is set as follows: AND $\{109.1k, 105.7k; 86.7k\}$, OR $\{83.6k, 85.9k; 262.5k\}$, IN1 $\{78.4k, 233.2k; 109.1k\}$ and IN2 $\{253.5k, 82.8k; 39.2k\}$. Similarly to Fig. 3a,b we also include the input/output mapping on the quantised, binary space. An input activates (deactivates) a specific memristor branch of the 1T1M array by getting LOW (HIGH), due to the use of pMOS as the transistors in 1T1M.

This example is showcasing clearly the reconfigurability of such systems and how they classify their inputs to a binary set of responses. Through this more detailed measurement approach, we can see more clearly the function-wise connection of the MCMTLG with the perceptron units and the ANNs. Similarly to the previous experiment with the 2-input MCMTLG, the 1T1M sub-circuits are conductive for LOW logic input and non-conductive for HIGH logic inputs, due to use of pMOS components for the memristive array.

B. 3-input & 4-input MCMTLG

For TL gates with larger input vector capabilities, thus greater fan-in, an increased number of 1T1M circuits is introduced to the MCMTLG's differential part. Through the employment of memristor memory cells, we are able to define the input weight through a single programmable resistive switch, hence eliminating large parallel pMOS networks for the differential part [25]. Hence improvements in speed and power can be observed [76], potentially closing the gap between TL technology and low-power computing systems at the edge. To support this concept through practical measurements of practically realized circuits MCMTLGs of higher input spaces were implemented and investigated. Fig.6a shows, a 3-input MCMTLG design, where a third 1T1M sub-circuit has been added to the input branch of the differential part and memristors are employed as the analogue weights at the input/bias binary signals, similar to the 2-input MCMTLG.

The measured results (Fig.6b,c) are extracted from a 3-element input vector and 1-element threshold setting of MCMTLG. The upper trace of Fig.3c shows the response for $\{M1, M2, M3; TH\} = \{31.5k\Omega, 30k\Omega, 28.2k\Omega; 68.2k\Omega\}$ memristive weight configuration which functions as a 3-input OR, where at least one of the input sub-circuit need to be conductive in order for the current comparison to result in an input-side winning node (majority-1 function). In Fig.6b the measured response for the canonical output (CA) while in Fig.6c the output of the isolation inverter connected to the CA output is shown. We can observe that during the equalization phase the voltage level or the isolation inverter is 0V while performing a full voltage swing to logic '1' when the corresponding CA output drops to logic '0'. The input vector and clock signal are shown in Fig.6d-g. In the case of memristor devices use in this configuration, similarly to the 2-input case described above, the same circuit could perform multiple functions just by reprogramming the threshold resistive value, thus configure differently the winning conditions of the current comparison performed.

Similarly to the operation of a perceptron or other simple ANN, we can observe that the memristive synaptic weights are responsible for the plane splitting of the 2D or 3D space. Through arbitrary reconfiguration of the memristive values, we are able to shift the decision boundary of neural decision-making functionality and thus alter the system of inequalities that the MCMTLG solves. The MCMTLG and TLGs, in general, are best suited for functions with high input space, thus represented by exploiting larger 1T1M arrays for the input and threshold differential part. Hence, our focus to experimentally test larger than 2- and 3-input MCMTLGs, and gather data from the gate-level testing of a memristively enhanced TL circuit.

In the example shown in Fig.7, a 4-input MCMTLG is tested, performing a MAJ-2 (majority-2) TL function, implemented by configuring the 1T1M TL arrays as follows: $\{M1, M2, M3, M4; TH\} = \{30k\Omega, 21.6k\Omega, 31.2k\Omega, 25.2k\Omega; 19.1k\Omega\}$. In Fig.7a the schematic of the circuit practically realized is shown, which include an additional 1T1M for the input network of the differential part. In Fig.7b-i the circuit response and the corresponding control signals (input vector and clock) are shown. It is worth noting that the C_{ab} (Fig.7h) and C_{ob} (Fig.7i) sensor outputs are the outputs of the isolation inverters, included into the sensor part by our approach here.

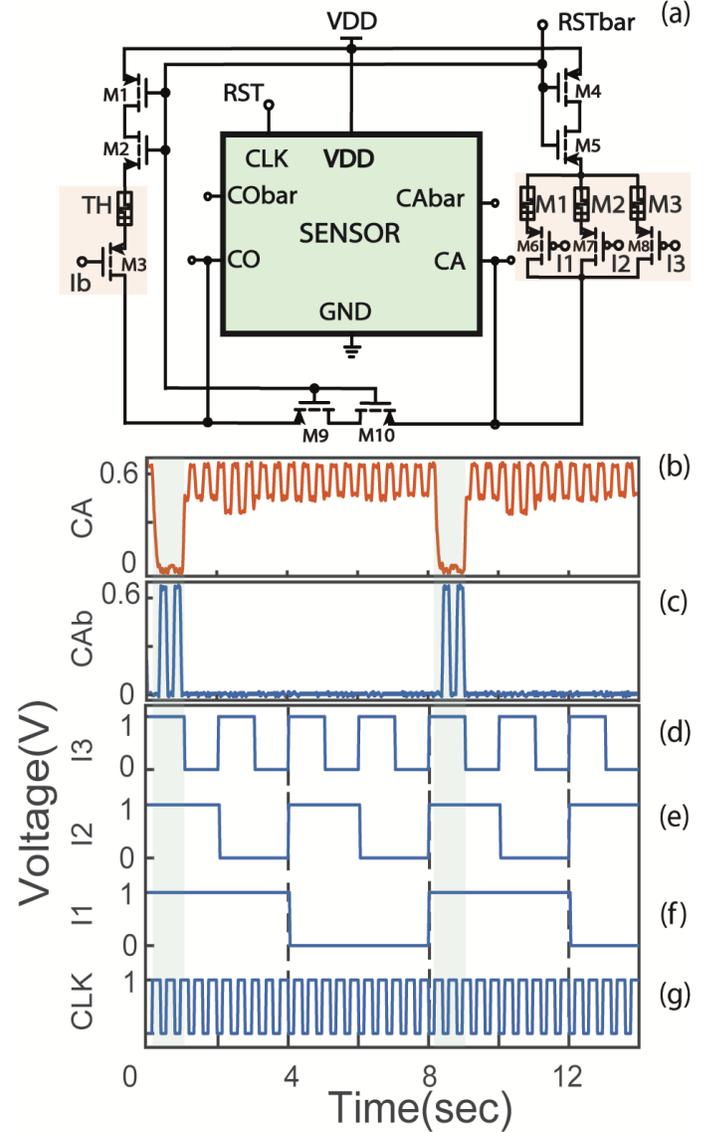


Fig. 6. Practical implementation of 3-input MCMTLG experiment. (a) Schematic of 3-input MCMTLG, where the sensor component now includes the CMOS latching element plus two additional restoration inverters per output. An example of how a 3D input space is split by the TLG. (b) Canonical (CA) and (c) complementary (CAB) output signals. (d-g) control signals (3-input vector $\{I1, I2, I3\}$ plus clock). The weight configuration for the OR3/NOR3 is $\{M1, M2, M3; TH\} = \{31.5k\Omega, 30k\Omega, 28.2k\Omega; 68.2k\Omega\}$.

We provide emphasis on the logic scaling capabilities of the memristor-based TLGs, including the proposed MCMTLG design presented in this work, as it can make this group of gates a promising candidate for advanced memory-centric

reconfigurable fabric implementation, where the functionality of a computational fabric is controlled by the ReRAM memory contents distributed into this sea of gates. The importance of logic scaling capabilities of said circuits has been explored previously [44], [47], [77], [78], i.e. the capability of TLGs to replace larger CMOS circuit with a single gate, that is why we considered an important parameter of our investigation the experimental demonstration of MCMTLGs with larger 1T1M arrays.

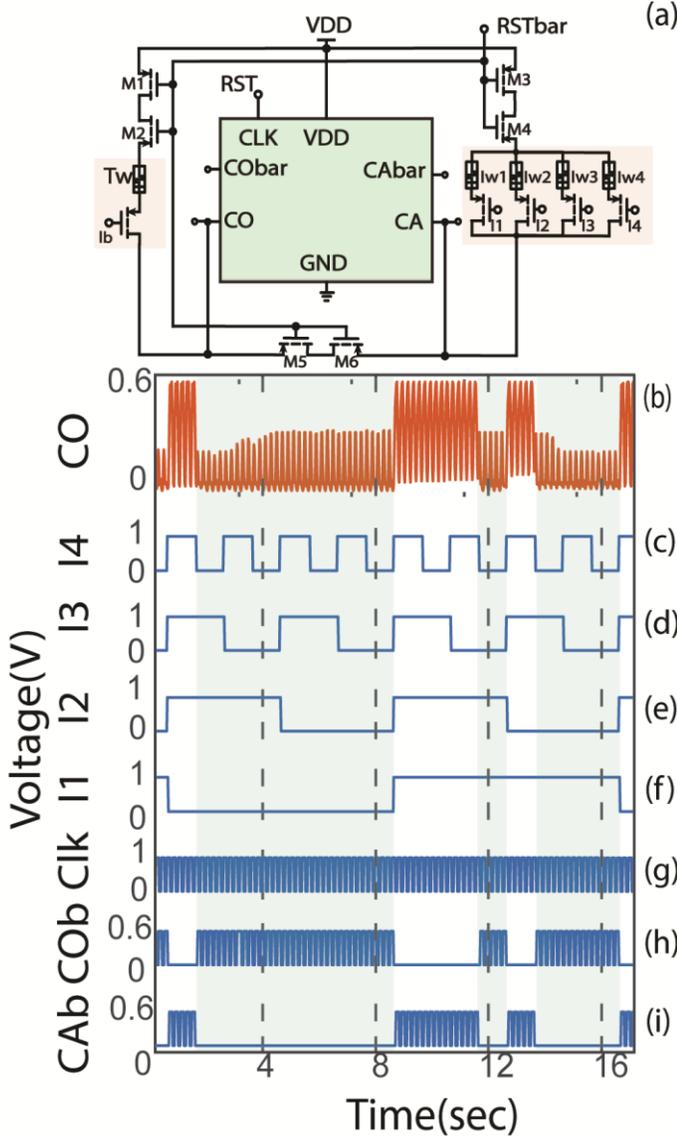


Fig. 7. Practical implementation of 4-input MCMTLG. (a) Schematic of the 4-input MCMTLG. (b) Circuit response measured at the complementary output (CO). (c-f) Input vector (I1, I2, I3, I4). (g) Clock signal. Additionally, to the analogous measured CO output of the sensor, the thresholded responses from the isolation inverters are also shown (h,i). The memristive weight configuration is $\{M1, M2, M3, M4; TH\} = \{30k\Omega, 21.6k\Omega, 31.2k\Omega, 25.2k\Omega; 19.1k\Omega\}$. This configuration of the MCMTLG performs the following Boolean function: $F = MAJ2(I1, I2, I3, I4)$, thus needing at least two conductive inputs for the total input current to be larger than the threshold current.

The larger differential arrays enable us to increase the fan-in capabilities of the gate, thus the complexity of linearly separable functions that we can represent on the memristive memory components is increasing. The replacement of multi-stage

CMOS logic, with an equivalent TLG circuit, can result in significant reduction in hardware cost, in power consumption and IC area as well as reduction of critical signal path length, thus even further improvements in performance and reduction in power and area [45].

C. Comparison with existing memristor-based TLG designs

The main contribution presented in this work is the practical implementation of said circuit using real memristors and thus the investigation of the feasibility of such circuits using state-of-the-art memristor technology. However, the MCMTLG circuits we have investigated are not the only embodiments of memristor-enhanced TL. A wealth of other designs have been proposed, albeit only examined through simulations, thus making it difficult to carry out an immediate comparison. In order to provide comparison data, we have modeled the MCMTLG in the industry-standard Cadence tool using TSMC's 65nm technology node for the CMOS transistors and our Verilog-A memristor model [79], (which is based on measurements taken from real memristor devices [79]). Hence, attempting to predict the performance of MCMTLG implemented in deep-submicron technologies and at the same time providing a better framework for comparison with existing memristor-based TLG designs. For the modeled MCMTLG we used memristors with values of $\{30k\Omega, 30k\Omega, 30k\Omega; 18k\Omega\}$, performing a majority-2 (MAJ-2) function, with voltage supply of 1V and clock period of 100ns. The widths of pMOS and nMOS devices were set to 400nm and 200nm, respectively. The power of MCMTLG was measured by calculating the product of $I \times V$ of the circuit's voltage supply, over a single clock period and for the case where all three 1T1M branches of the input array are conductive (input case: logic "111"). The delay was measured for the start of the evaluation phase until the voltage difference between the two complementary outputs, CO and CA, of the sensor, reached $V_{DD}/2 = 500mV$. We have tested for worst case scenario vis-a-vis memristor resistive states (exactly two inputs are active, leading to a close comparison between 15k Ω and 18k Ω composite memristances).

Next, we choose few memristive TLG designs representing major directions in circuit implementation from the literature. A performance comparison across different metrics is shown in Table I. The Memristor-based TL (MTL) [59] design was one of the first TLGs that made use of memristors as its input weights. In MTL the memristive weights are isolated from the actual thresholding units (a cascade of inverters) through the use of current mirrors. The Resistive TL (RTL) [62] is similar to the MTL, but with the simplification that the memristive network is directly connected to the output inverter. Additionally, the RTL gate uses a type of ratioed memristive network, thus a network with pull-up and pull-down branches (for the specific RTL gate a single pull-down memristor), thus enhancing the reconfiguration capabilities of the gate. The CMMTL [41] design was one of the first differential TLG concept that used memristors. In terms of design, CMMTL is similar to the proposed MCMTLG implementation, having its differential part separated from the sensor part. CMMTL tested provided a glimpse regarding some of the advantages the differential implementations have over non-differential TL. More specifically, CMMTL showcased better energy and delay metrics over the MTL implementation [41]. For the comparison,

the first order Threshold Function memristive Threshold Logic Gate (1-TF mTG) implementation, a state-of-art differential current-mode memristive TLG [42], is used as well. In this work the TLGs are optimized for minimal transistor count and can achieve very low power consumption and delay. In terms of design, the 1-TF mTG incorporates in a way the differential part into the sensor part, with the differential arrays being part of the pull-up network of the gate. The design is very interesting and is similar to the work of Bobba et al. [26], which was one of the first to proposed CMOS-based CM-TLGs.

TABLE I
COMPARISON WITH EXISTING MEMRISTIVE TLGS

mTLG designs	Power	Delay	#Devices
CMMTL [41]	118.22uW	0.44ns	24
MTL [52]	35uW	6.1ns	24
RTL [62]	9.2uW	0.45us	5
1-TF mTG [42]	0.27uW	0.195ns	15
Proposed MCMTLG	27.2uW	0.14ns	24

For our comparison, we use published data that support the metrics of power consumption and delay. For the case of CMMTL, we used data for the delay metric is measured by [41] using 45nm Berkley's Predictive Technology Model (PTM) technology node and for 3-input gate. For the power estimation of the CMMTL we could not gather relevant data from [41], but in the comparative study by [53] an estimation is provided of power supply with 0.25um TSMC technology node and for 2-input case study. The MTL data were gathered from [59], using 45nm PTM technology node for a 3-input case study. The RTL metrics were gathered by [62], using a 0.25um TSMC technology node and the data refers to 2-input gates. The 1-TF mTG power and delay was measured in [42] using 45nm PTM technology node and the used values are for an indicative 3-input gate. CMMTL, MTL and RTL used HP memristor model [41], [53], [59] for their simulations, while 1-TF mTG simulations were based on VTEAM memristor model [42]. From TABLE I we observe that our implementation offers competitive power dissipation. In terms of device count (transistors and memristors), MCMTLG and CMMTL have the larger count of devices, while the much simpler design of RMTL has the least components. But in terms of delay, MCMTLG is comparable with the state-of-the-art 1-TF mTG implementation. We further observe that in general differential implementations have smaller delay than non-differential ones.

VI. DISCUSSION & CONCLUSIONS

In this work, we have presented a physical implementation of a CM-TLG circuit which uses reconfigurable memristive loads as analogue weights. Through the presented experimental results we have practically demonstrated the functionality of 2-, 3-, and 4-input memristor-based TLGs, showcasing the memristor-enabled reconfigurability of the base design for all validated cases. Furthermore, we have investigated how memristor-based TLGs behave when used outside their intended, digital-input operating regime. Experiments show the decision boundary shapes, which approximate bevelled L-shapes with lines parallel to the input voltage axes. Further investigation through simulations confirms these shapes and shows graphically how altering the resistive states of the

memristive devices affects the specific decision boundary locations (most notably which points in the binary-quantised input space lie on the 'output = 1' side of the boundary or the 'output=0'). Finally, we implement and simulate a 3-input TLG in a TSMC 65nm technology for the purposes of comparison vs state-of-art. To that purpose, we have utilised our own memristor model which takes into account the non-linearity of the memristive device IV (a typical feature of many technologies [11]). Results testify to the robustness of the TLG concept by confirming no perturbation of functionality and power/delay figures comparable, and indeed competitive, vs. state-of-art. Our work thus provides experimental backing to a considerable body of literature where simulation work indicates the potential for highly energy- and area-efficient TL implementations exploiting memristive devices.

Finally, the present work suggests some routes for further investigation. Notably, we would expect the capability of memristors to support fine tuning of their resistances to become particularly valuable in TLGs of even higher dimensionality. In a 5-input TLG, the number of possible points increases to 32 whilst the number of memristive devices only rises to 6. It would be useful to investigate how the number of possible majority gates implementable with n inputs increases vs the number of available memristors ($n+1$), and computing a 'logic density' metric. We conjecture that higher n will lead to a higher 'logic density' burden, which eventually reaches the maximum number of resolvable states attainable by the memristor. How this limits practical performance remains to be revealed. Another avenue of investigation pertains to linking multiple TLGs together in larger combinatorial blocks and understanding, e.g. how the memristor resistance-dependent delays may affect overall timing constraints (particularly in domino logic-type systems).

REFERENCES

- [1] D. E. Nikonov and I. A. Young, "Overview of beyond-CMOS devices and a uniform methodology for their benchmarking," *Proc. IEEE*, vol. 101, no. 12, pp. 2498–2533, 2013.
- [2] S. Hamdioui, M. Taouil, H. A. Du Nguyen, A. Haron, L. Xie, and K. Bertels, "Memristor: The enabler of computation-in-memory architecture for big-data," *2015 Int. Conf. Memristive Syst. MEMRISYS 2015*, pp. 9–11, 2016.
- [3] S. A. McKee, "Reflections on the Memory Wall," *Proc. 1st Conf. Comput. Front.*, pp. 162–167, 2004.
- [4] M. Wilkes, "The memory wall and the CMOS end-point," *ACM SIGARCH Comput. Archit. News*, pp. 1994–1996, 1995.
- [5] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 459, no. 7250, pp. 1154–1154, 2009.
- [6] A. Khiat, P. Ayliffe, and T. Prodromakis, "High Density Crossbar Arrays with Sub- 15 nm Single Cells via Liffoff Process Only," *Sci. Rep.*, vol. 6, no. 1, p. 32614, 2016.
- [7] S. Stathopoulos *et al.*, "Multibit memory operation of metal-oxide Bilayer memristors," *Sci. Rep.*, vol. 7, no. 1, 2017.
- [8] L. Chua, "If it's pinched it's a memristor," *Semicond. Sci. Technol.*, vol. 29, no. 10, pp. 1–42, 2014.
- [9] L. Chua, "Everything you wish to know about memristors but are afraid to ask," *Radioengineering*, vol. 24, no. 2, pp. 319–368, 2015.
- [10] J. Joshua Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," *Nat. Nanotechnol.*, vol. 8, 2013.
- [11] F. Merrikh-Bayat, M. Prezioso, X. Guo, B. Hoskins, D. B. Strukov, and K. K. Likharev, "Memory Technologies for Neural Networks," *2015 IEEE 7th Int. Mem. Work. IMW 2015*, 2015.
- [12] D. Ielmini and H.-S. P. Wong, "In-memory computing with resistive switching devices," *Nat. Electron.*, vol. 1, no. 6, pp. 333–343, 2018.
- [13] Y. Guo, X. Wang, and Z. Zeng, "A Compact memristor-CMOS hybrid

- Look-up-table Design and Potential Application in FPGA,” *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 36, no. 12, pp. 1–1, 2017.
- [14] S. Choi, J. H. Shin, J. Lee, P. Sheridan, and W. D. Lu, “Experimental Demonstration of Feature Extraction and Dimensionality Reduction Using Memristor Networks,” *Nano Lett.*, vol. 17, no. 5, pp. 3113–3118, 2017.
- [15] M. Tarkhan and M. Maymandi-Nejad, “Design of a memristor based fuzzy processor,” *AEU - Int. J. Electron. Commun.*, vol. 84, no. August 2017, pp. 331–341, 2018.
- [16] S. Hamdioui *et al.*, “Memristor Based Computation-in-Memory Architecture for Data-Intensive Applications,” in *Design, Automation & Test in Europe Conference & Exhibition (DATE) 2015*, 2015.
- [17] H. Wu, P. Yao, B. Gao, and H. Qian, “Multiplication on the edge,” *Nat. Electron.*, vol. 1, no. 1, pp. 8–9, 2018.
- [18] O. Krestinskaya, I. Dolzhikova, and A. P. James, “Hierarchical Temporal Memory using Memristor Networks: A Survey,” pp. 1–15, 2018.
- [19] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943.
- [20] M. Abadi *et al.*, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” *arXiv:1603.04467 [cs.DC]*, 2016.
- [21] M. Vestias and H. Neto, “Trends of CPU, GPU and FPGA for high-performance computing,” *Conf. Dig. - 24th Int. Conf. F. Program. Log. Appl. FPL 2014*, 2014.
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2012.
- [23] F. M. Bayat, M. Prezioso, B. Chakrabarti, I. Kataeva, and D. Strukov, “Memristor-based perceptron classifier: Increasing complexity and coping with imperfect hardware,” in *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*, 2017, pp. 549–554.
- [24] V. Beiu, J. M. Quintana, and M. J. Avedillo, “VLSI implementations of threshold logic-a comprehensive survey,” *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1217–43, 2003.
- [25] S. Leshner, N. Kulkarni, S. Vrudhula, and K. Berezowski, “Design of a robust, high performance standard cell threshold logic family for DSM technology,” in *Proceedings of the International Conference on Microelectronics, ICM*, 2010, pp. 52–55.
- [26] S. Bobba and N. Hajj, “Current-Mode Threshold Logic Gates,” in *Computer Design, 2000. Proceedings. 2000 International Conference on*, 2000, pp. 235–240.
- [27] T. Tran, A. Rothenbuhler, E. H. B. Smith, V. Saxena, and K. A. Campbell, “Reconfigurable Threshold Logic Gates using memristive devices,” *2012 IEEE Subthreshold Microelectron. Conf. SubVT 2012*, pp. 174–193, 2012.
- [28] A. P. James, A. Pachentavida, and S. Sugathan, “Edge detection using resistive threshold logic networks with CMOS flash memories,” *Int. J. Intell. Comput. Cybern.*, vol. 7, no. 1, pp. 79–94, 2014.
- [29] O. Krestinskaya, A. Kumar Maan, and A. Pappachen James, “Programmable Memristive Threshold Logic Gate Array,” *arXiv:1809.00419v1 [cs.ET]*, 2018.
- [30] L. Gao, F. Alibart, and D. B. Strukov, “Programmable CMOS/memristor threshold logic,” *IEEE Trans. Nanotechnol.*, vol. 12, no. 2, pp. 115–119, 2013.
- [31] A. H. Edwards, H. J. Barnaby, K. A. Campbell, M. N. Kozicki, W. Liu, and M. J. Marinella, “Reconfigurable memristive device technologies,” *Proc. IEEE*, vol. 103, no. 7, pp. 1004–1033, 2015.
- [32] Y. Li, Z. Wang, R. Midya, Q. Xia, and J. Joshua Yang, “Review of memristor devices in neuromorphic computing: Materials sciences and device challenges,” *J. Phys. D. Appl. Phys.*, vol. 51, no. 50, 2018.
- [33] M. Le Gallo *et al.*, “Mixed-Precision In-Memory Computing,” *arXiv:1701.04279v4 [cs.ET]*, pp. 1–11, 2017.
- [34] G. W. Burr *et al.*, “Neuromorphic computing using non-volatile memory,” *Adv. Phys. X*, vol. 2, no. 1, pp. 89–124, 2017.
- [35] S. Ambrogio *et al.*, “Equivalent-accuracy accelerated neural-network training using analogue memory,” *Nature*, vol. 558, no. 7708, pp. 60–67, 2018.
- [36] N. S. Nukala, N. Kulkarni, and S. Vrudhula, “Spintronic Threshold Logic Array (STLA) - A compact, low leakage, non-volatile gate array architecture,” *J. Parallel Distrib. Comput.*, 2014.
- [37] Z. He and D. Fan, “Energy efficient reconfigurable threshold logic circuit with spintronic devices,” *IEEE Trans. Emerg. Top. Comput.*, vol. 5, no. 2, pp. 223–237, 2017.
- [38] F. Alibart, L. Gao, and D. B. Strukov, “A Reconfigurable FIR Filter with Memristor-Based Weights,” *arXiv:1608.05445 [cs.ET]*, 2016.
- [39] V. Beiu, J. M. Quintana, M. J. Avedillo, and R. Andonie, “Differential implementations of threshold logic gates,” *SCS 2003 - Int. Symp. Signals, Circuits Syst. Proc.*, vol. 2, pp. 489–492, 2003.
- [40] V. Beiu, J. M. Quintana, M. J. Avedillo, M. S. I. E. Cica, and A. Reina, “Threshold Logic: From Vacuum Tubes to Nanoelectronics,” in *2003 46th Midwest Symposium on Circuits and Systems*, 2003, pp. 930–935.
- [41] C. B. Dara, T. Haniotakis, and S. Tragoudas, “Low Power and High Speed Current-Mode Memristor-Based TLGs,” in *IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS)*, 2013, pp. 89–94.
- [42] S. N. Mozaffari and S. Tragoudas, “Maximizing the Number of Threshold Logic Functions Using Resistive Memory,” *IEEE Trans. Nanotechnol.*, vol. 17, no. 5, pp. 897–905, 2018.
- [43] C. B. Dara, T. Haniotakis, and S. Tragoudas, “Delay Analysis for Current Mode Threshold Logic Gate Designs,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 25, no. 3, pp. 1063–1071, 2017.
- [44] S. N. Mozaffari, S. Tragoudas, and T. Haniotakis, “A Generalized Approach to Implement Efficient CMOS-Based Threshold Logic Functions,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 65, no. 3, pp. 946–959, 2018.
- [45] S. Leshner, “Modeling and Implementation of Threshold Logic Circuits and Architectures,” Arizona State University, PhD Thesis, 2010.
- [46] A. P. James, “Memristor Threshold Logic: An Overview to Challenges and Applications,” in *International Conference on Contemporary Computing and Informatics*, 2016, pp. 13–16.
- [47] S. N. Mozaffari, S. Tragoudas, and T. Haniotakis, “Reducing power, area, and delay of threshold logic gates considering non-integer weights,” in *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*, 2017, pp. 3–6.
- [48] V. Seshadri *et al.*, “Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology,” in *In Proceedings of Annual IEEE/ACM International Symposium on Microarchitecture*, 2017.
- [49] H. Inokawa, A. Fujiwara, and Y. Takahashi, “A Multiple-Valued Logic and Memory With Combined Single-Electron and Metal-Oxide-Semiconductor Transistors,” *IEEE Trans. Electron Devices*, vol. 50, no. 2, 2003.
- [50] S. M. Mirhoseini, M. J. Sharifi, and D. Bahrepour, “New RTD-based general threshold gate topologies and application to three-input XOR logic gates,” *J. Electr. Comput. Eng.*, vol. 2010, no. 1, pp. 1–5, 2010.
- [51] H. Pettenghi, M. J. Avedillo, and J. M. Quintana, “Using multi-threshold threshold gates in RTD-based logic design: A case study,” *Microelectronics J.*, vol. 39, no. 2, pp. 241–247, 2008.
- [52] J. Rajendran, H. Manem, R. Karri, and G. S. Rose, “Memristor based programmable threshold logic array,” *Proc. 2010 IEEE/ACM Int. Symp. Nanoscale Archit. NANOARCH 2010*, pp. 5–10, 2010.
- [53] A. Kumar Maan, D. Anirudhan Jayadevi, and A. Pappachen James, “A Survey of Memristive Threshold Logic Circuits,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 8, pp. 1734–1746, 2016.
- [54] N. Kulkarni, J. Yang, J. S. Seo, and S. Vrudhula, “Reducing Power, Leakage, and Area of Standard-Cell ASICs Using Threshold Logic Flip-Flops,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 24, no. 9, pp. 2873–2886, 2016.
- [55] J. Yang, N. Kulkarni, S. Yu, and S. Vrudhula, “Integration of threshold logic gates with RRAM devices for energy efficient and robust operation,” *Proc. 2014 IEEE/ACM Int. Symp. Nanoscale Archit. NANOARCH 2014*, no. 3, pp. 39–44, 2014.
- [56] S. Vrudhula, N. Kulkarni, and J. Yang, “Design of threshold logic gates using emerging devices,” *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. 2015–July, pp. 373–376, 2015.
- [57] S. Tatapudi and V. Beiu, “Split-Precharge Differential Noise-Immune Threshold Logic Gate (SPD-NTL),” in *In: Mira J., Álvarez J.R. (eds) Artificial Neural Nets Problem Solving Methods. IWANN 2003. Lecture Notes in Computer Science, vol 2687. Springer, Heidelberg*, 2003.
- [58] J. Y. Lee and S. N. Hwang, “A high-gain boost converter using voltage-stacking cell,” *Trans. Korean Inst. Electr. Eng.*, vol. 57, no. 6, pp. 982–984, 2008.
- [59] J. Rajendran, H. Manem, R. Karri, and G. S. Rose, “An Energy-Efficient Memristive Threshold Logic Circuit,” *IEEE Trans. Comput.*, vol. 61, no. 4, pp. 474–487, 2012.
- [60] A. P. James, D. S. Kumar, and A. Ajayan, “Threshold Logic Computing: Memristive-CMOS Circuits for Fast Fourier Transform and Vedic Multiplication,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 23, no. 11, pp. 2690–2694, 2015.
- [61] A. Kumar Maan and A. Pappachen James, “Voltage Controlled

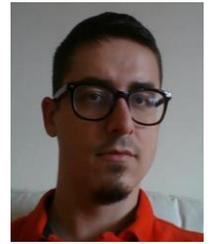
Memristor Threshold Logic Gates,” *IEEE Asia Pacific Conf. Circuits Syst. 2016*, 2016.

- [62] A. P. James, L. R. V. J. Francis, and D. Kumar, “Resistive Threshold Logic,” *arXiv:1208.0090v1 [cs.ET]*, pp. 1–12, 2013.
- [63] A. K. Maan, D. S. Kumar, S. Sugathan, and A. P. James, “Memristive Threshold Logic Circuit Design of Fast Moving Object Detection,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 23, no. 10, pp. 2337–2341, 2015.
- [64] Q. Zhu *et al.*, “A 3D-stacked logic-in-memory accelerator for application-specific data intensive computing,” *2013 IEEE Int. 3D Syst. Integr. Conf. 3DIC 2013*, 2013.
- [65] A. K. Maan, D. S. Kumar, and A. P. James, “Memristive threshold logic face recognition,” *Energy Procedia*, vol. 41, no. C, pp. 98–103, 2014.
- [66] R. Zhang and M. Kaneko, “A 16-valued logic FPGA architecture employing analog memory circuit,” *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. 2016–July, pp. 718–721, 2016.
- [67] Y. C. Chen, W. Zhang, and H. Li, “A look up table design with 3D bipolar RRAMs,” *Proc. Asia South Pacific Des. Autom. Conf. ASP-DAC*, 2012.
- [68] T. N. Kumar, H. A. F. Almurib, and F. Lombardi, “A Novel Design of a Memristor-Based Look-Up Table (LUT) For FPGA,” in *Circuits and Systems (APCCAS), 2014 IEEE Asia Pacific Conference on*, 2014.
- [69] M. Teimoori, A. Ahmadi, S. Alirezadee, and M. Ahmadi, “A novel hybrid CMOS-memristor logic circuit using Memristor Ratioed Logic,” *Can. Conf. Electr. Comput. Eng.*, vol. 2016–Octob, no. 1, pp. 1–4, 2016.
- [70] M. B. A. A. S. Emara, A.H. Madian, H.H. Amer, S.H Amer, “Testing of Memristor Ratioed Logic (MRL) XOR Gate,” *IEEE ICM 2016*, 2016.
- [71] S. Kvatinisky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, and E. G. Friedman, “MRL - Memristor Ratioed Logic,” in *International Workshop on Cellular Nanoscale Networks and their Applications*, 2012.
- [72] J. A. Hidalgo-Lopez, J. C. Tejero, J. Fernandez, and A. Gago, “New types of digital comparators,” *Proc. 1995 IEEE Int. Symp. Circuits Syst. 95. Part 3 (of 3)*, vol. 1, pp. 29–32, 1995.
- [73] J. F. Ramos, J. A. H. López, M. J. Martin, J. C. Tejero, and A. Gago, “A threshold logic gate based on clocked coupled inverters,” *Int. J. Electron.*, vol. 84, no. 4, pp. 371–382, 1998.
- [74] S. Pi *et al.*, “Memristor Crossbars with 4.5 Terabits-per-Inch-Square Density and Two Nanometer Dimension,” *arXiv:1804.09848 [cond-mat.mes-hall]*, pp. 1–14, 2018.
- [75] G. Papandroulidakis, A. Khiat, A. Serb, S. Stathopoulos, L. Michalas, and T. Prodromakis, “Metal Oxide-enabled Reconfigurable Memristive Threshold Logic Gates,” in *IEEE International Symposium on Circuits and Systems (ISCAS) 2018*, 2018.
- [76] S. N. Mozaffari, S. Tragoudas, and T. Haniotakis, “A Generalized Approach to Implement Efficient CMOS-Based Threshold Logic Functions,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 65, no. 3, pp. 946–959, 2018.
- [77] A. Wagle, J. Yang, A. Dengi, and S. Vrudhula, “FPGAs with Reconfigurable Threshold Logic Gates for Improved Performance, Power and Area,” *2018 Int. Conf. Field-Programmable Log. Appl.*, pp. 256–259, 2018.
- [78] O. Krestinskaya and A. P. James, “Approximate Probabilistic Neural Networks with Gated Threshold Logic,” *arXiv:1808.00733 [cs.ET]*, pp. 10–13, 2018.
- [79] I. Messaris, A. Serb, S. Stathopoulos, A. Khiat, S. Nikolaidis, and T. Prodromakis, “A Data-Driven Verilog-A ReRAM Model,” *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. TBA, no. TBA, pp. 1–12, 2018.

Georgios Papandroulidakis received his M.Eng. degree in electrical and computer engineering (ECE) from the Democritus University of Thrace (DUTH), Greece, in 2015, and his M.Sc. degree in ECE from DUTH, Greece, in 2017. He is currently pursuing his Ph.D. studies at University of Southampton in Electronic Materials & Devices (EMD) research group. His work is focused on reconfigurable hybrid CMOS/memristor circuits and systems for logic and memory applications.



Alexander Serb (M11) has graduated from Imperial College in 2009 as a biomedical engineer and later obtained his PhD from Imperial College in 2013 in electronics engineering. Since then he has been a research fellow in prof. Prodromakis' group, working on memristors technology characterisation and application development. His research interests include circuit design, computational neuroscience and biology-machine interfaces.



Ali Khiat is an experimental officer within Electronic Materials & Devices Research Group - the Zepler Institute and Southampton Nanofabrication Centre at the University of Southampton. Previously he was Research Associate within the Centre for Bio-inspired Technology at Imperial College London (UK). He has also worked at Delft University of Technology (The Netherlands) in different projects related to micro-actuators, micro-sensors and carbon nanotubes nano-switches. He obtained PhD from Compiegne University of Technology (France) in advanced mechanics and applications, where he developed a fibre optic sensor. He was also research visitor to the Institute of Microtechnology - Braunschweig University of Technology (Germany). He obtained his engineering degree from USTHB University (Algeria).



Geoff Merrett (GSM06-M09) received the B.Eng. degree (Hons.) in electronic engineering and the Ph.D. degree from the University of Southampton, Southampton, U.K., in 2004 and 2009, respectively. He is currently an Associate Professor and Head of the Centre for IoT and Pervasive Systems at the University of Southampton. His research are energy-efficient computing, particularly around system-level management of many-core and battery-free systems. He has published over 150 articles in journals/conferences in these areas, and is co-editor of the “Many-Core Computing: Hardware and Software” (IET Press, 2019).



Themis Prodromakis (SM08) is a Professor of Nanotechnology and Head of the Electronic Materials and Devices Research Group in the Zepler Institute, University of Southampton. He is recognised as a pioneer of metal-oxide Resistive Random-Access Memory technologies and is leading an interdisciplinary team comprising 20 researchers with expertise ranging from materials process development to electron devices and circuits and systems for embedded applications.



He holds an EPSRC Fellowship, a Royal Society Industry Fellowship and is a Visiting Professor at the Department of Microelectronics and Nanoelectronics at Tsinghua University, CN and Honorary Fellow at Imperial College London. He previously held a Corrigan Fellowship in Nanoscale Technology and Science, funded by the Corrigan Foundation and LSI Inc., within the Imperial College and a Lindemann Trust Visiting Fellowship in EECS UC Berkley.

His background is in Electron Devices and nanofabrication techniques, with his research being focused on bio-inspired devices for advanced computing architectures and biomedical applications.