Enhanced Scouting Logic
A Robust Memristive Logic Design Scheme

Yu, Jintao; Du Nguyen, Hoang Anh; Abu Lebdeh, Muath; Taouil, Mottaqiallah; Hamdioui, Said

DOI
10.1109/NANOARCH47378.2019.181296

Publication date
2020

Document Version
Accepted author manuscript

Published in
NANOARCH 2019 - 15th IEEE/ACM International Symposium on Nanoscale Architectures, Proceedings

Citation (APA)
Yu, J., Du Nguyen, H. A., Abu Lebdeh, M., Taouil, M., & Hamdioui, S. (2020). Enhanced Scouting Logic: A Robust Memristive Logic Design Scheme. In *NANOARCH 2019 - 15th IEEE/ACM International Symposium on Nanoscale Architectures, Proceedings* (pp. 1-6). Article 9073668 (NANOARCH 2019 - 15th IEEE/ACM International Symposium on Nanoscale Architectures, Proceedings). IEEE. https://doi.org/10.1109/NANOARCH47378.2019.181296

Enhanced Scouting Logic: A Robust Memristive Logic Design Scheme

Jintao Yu Hoang Anh Du Nguyen Muath Abu Lebdeh Mottaqiallah Taouil Said Hamdioui
Laboratory of Computer Engineering, Delft University of Technology
Delft, the Netherlands
{J.Yu-1, H.A.DuNguyen, M.F.M.AbuLebdeh, M.Taouil, S.Hamdioui}@tudelft.nl

***Abstract*—Memristive devices have the potential to reduce the memory access bottleneck in conventional computer architectures. However, memristive devices also suffer from low endurance and large resistance variation. To address these problems, we present a robust logic scheme named Enhanced Scouting Logic (ESL). It produces logic operation results within the peripheral circuit of the memory array. During the execution of logic operations, the resistance states of memristive devices do not change and hence do not affect the memristor lifetime. ESL senses the resistance of input memristive devices via two different paths when different operations such as AND and OR are performed. These different paths guarantee the operation correctness even under large resistance variations. We verified ESL using SPICE simulations and Monte Carlo analysis. Our simulation results show that ESL is more robust as compared with state-of-the-art logic schemes.**

I. Introduction

Today's big-data applications demand high performance and high energy efficiency, challenging both CMOS technology and conventional computer architectures [1]. In CMOS technology, the energy problem worsens as the leakage power increases with downscaling [2]. In conventional CPU-centric architectures, the time and energy spent in accessing the main memory are significant; it can even exceed those of computations [3]. Therefore, new solutions are needed to alleviate the architecture and technology challenges. Emerging memristive devices have the potential to reduce static power due to their non-volatility. They also enable computation-in-memory as they can be used for both computing and storage [4]. Preliminary researches have exhibited the huge potential of memristive computing systems [5], [6]. Nevertheless, memristive devices also face multiple challenges. For instance, RRAM suffers from large resistance variation and low endurance; PCRAM requires long write time and high write energy; STT-MRAM has relatively poor process compatibility with mainstream silicon CMOS technology, which leads to a high manufacture cost [7], [8].

Most memristive logic schemes, such as Snider [9] and IMP [10], generate the results in the form of resistance states in the array. Therefore, they program the memristive devices frequently, hence reducing the device endurance. To avoid this problem, other schemes, such as Pinatubo [5] and Scouting Logic [11], generate the results in the periphery without switching the resistance states in the array; e.g., the results are produced as voltages at the output of sense amplifiers. These logic schemes do not affect the endurance, and hence, they maintain a longer lifetime. However, these schemes may not guarantee correctness when the input memristive devices exhibit large resistance variation. As far as we know, state-of-the-art logic schemes are not able to address both endurance and variation problems.

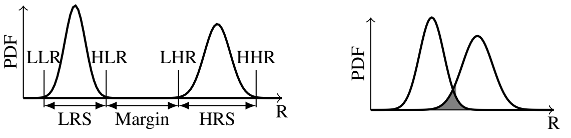
In this paper, we propose enhanced scouting logic (ESL), a logic scheme that improves Scouting Logic [11] by addressing the two major challenges when performing in-memory logic operations using RRAM; namely, the low endurance and the high resistance variation. Similarly as in Pinatubo and Scouting Logic, the proposed solution addresses the low endurance problem by producing the results in the memory periphery without changing the resistance states of the involved memory cells. It also addresses the high resistance variation problem by using 2T1R cells allowing two separate reading paths for AND and OR operations. The main contributions of this paper are:

- It reviews existing logic schemes that produce the outputs of logic operations within the memory periphery. Our results show that these schemes were not properly evaluated against realistic resistance variations of the RRAM devices.
- It proposes a memristive logic scheme that is resilient to resistance variation without affecting the memory endurance.
- It validates the proposed scheme and compares the scheme with the state-of-the-art using SPICE simulation and Monte Carlo analysis.

This paper is organized as follows. Section II and Section III review the resistance variation of RRAM devices and the logic styles that perform in memory peripheral, respectively. Subsequently, Section IV presents the working principle and implementation details of ESL. Next, Section V verifies ESL using SPICE simulation. In addition, Section V applies Monte Carlo simulation on ESL and other logic schemes. Finally, Section VI concludes the paper.

II. Resistance Variation in RRAM Devices

A typical RRAM device has two stable resistance states; i.e., a low resistance state (LRS) and a high resistance sate (HRS). The SET voltage and the RESET voltage are applied across the RRAM device to switch its resistance state to LRS and to HRS, respectively. The resistance (either LRS or HRS) of RRAM



(a) Separated ranges (b) Overlapped ranges

Fig. 1. Typical resistance distribution of LRS and HRS ranges.

TABLE I
RESISTANCE RANGES OF RECENT RRAM DEVICES

Ref.	Material	LRS range	$\frac{\text{HLR}}{\text{LLR}}$	HRS range	$\frac{\text{HHR}}{\text{LHR}}$	Margin
[15]	Pt-AlO$_y$/HfO$_x$-TiN	200-1k	5	80k-2M	25	80x
[16]	Cu-MgO-Ru	400-600	1.5	4k-50k	12.5	8x
[17]	Ti-TiN-HfO$_x$	300-600	2	3k-20k	6.6	10x
[12]	HfO$_2$-Hf	30k-80k	2.6	300k-1M	3.3	4x
[18]	Au-SiO$_x$-Mo	200-350	1.8	2k-10k	5	6x
[19]	Ti-SiO$_x$-C	200-350	1.8	2k-10k	5	6x
[20]	Ti/Ta-TMO$_x$	10k-50k	5	500k-500M	1000	10x

devices may vary as a result of the cycle-to-cycle (C2C) and device-to-device (D2D) variations [12]. The C2C and D2D variations have similar distributions [12] and can affect the correctness of logic operations. Due to their similarity, we do not distinguish between the C2C and D2D variations.

Fantini *et al.* observed that the distributions of LRS and HRS approximately follow a lognormal distribution [13], as shown in Fig. 1. The x axis represents the resistance in log scale while the y axis represents probability density function (PDF). The lower and upper bounds of LRS are named low low resistance (LLR) and high low resistance (HLR), respectively. Similarly, the lower and upper bounds of HRS are referred to as low high resistance (LHR) and high high resistance (HHR), respectively. We are particularly interested in the devices that have separate LRS and HRS ranges, i.e., LHR is larger than HLR, as shown in Fig. 1a. We refer to the ratio between LHR and HLR as *margin*. If LHR is smaller than HLR (i.e., negative margin) as shown in Fig. 1b, the proposed logic schemes will not work properly.

The ideal RRAM device should have small LRS and HRS ranges (i.e., narrow distributions of LRS and HRS) and a large margin; however, this is usually not the case. The LRS and HRS ranges can be estimated based on their distribution. For the LRS of an HfO$_2$ RRAM, $\sigma/\log_{10}\mu$ is between 0.2% and 10% [13]. As a result, the ratio between HLR and LLR (HLR/LLR) varies from 1.2 to 1000 considering a 3σ population. For example, when $\sigma/\log_{10}\mu = 2\%$ and $\mu = 10\,\text{k}\Omega$, the range of LRS is between $6.9\,\text{k}\Omega$ and $17\,\text{k}\Omega$ and HLR/LLR$= 2.5$.

Grossi *et al.* assume that LRS variations can be represented by a natural distribution. They summarize the RRAM's resistance variation based on seven publications [14]. In most cases, σ/μ of LRS resistance is larger than 10%. In some cases, this value is nearly 100%. Considering a 3σ population, we observe that HLR/LLR> 1.9. Note that the HRS range is even larger, but less important as it affects the operations less. This will be more clear in the next sections.

Table I shows the LRS and HRS ranges of recently proposed RRAM devices and the materials they are made of. The last column shows the margin between the two resistance states. The table shows that HLR/LLR$\in (1.5, 5)$ and $margin \in (4, 80)$. This indicates that LRS and HRS both have large variations, but also that a wide gap exists between them.

III. BULK BITWISE BOOLEAN LOGIC SCHEMES

Seshadri *et al.* was the first to propose a mechanism to perform *bulk* bitwise operations using specially designed peripheral circuits of memories [21]. These bitwise operations are performed on large vectors [31]. This idea inspired many researchers, including the usage of novel devices such as memristive devices. They are summarized in Table II.

The table contains for each reference the scheme name, used memory technology, the operations that are supported, whether variations are considered, and the worst case latency of the supported operations. Some articles target general non-volatile memories or memristive devices [5], [11], [25] while some other schemes work specifically for a particular memory such as STT-MRAM [26]–[29]. ESL, added to the last row of the table, can be applied to any type of memristive devices. However, it is most useful for devices with large resistance variations such as RRAM.

The fourth column of the table shows the different Boolean operations that are supported in the articles. Some schemes [26], [28], [30] support NAND/NOR operations by selecting the sense amplifier's $\overline{OUT}$ signal during AND/OR. This is easy to implement in all schemes and not the focus of this paper. Therefore, such operations are omitted in the table for simplicity.

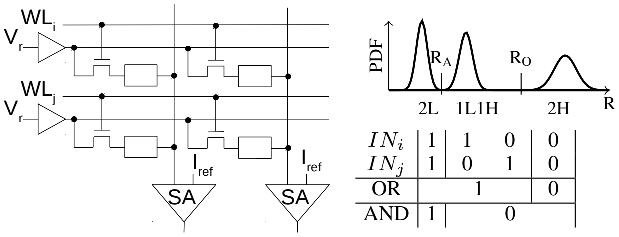
We are particularly interested in the fifth column, i.e., whether resistance variations have been considered. This is however not relevant for DRAM and SRAM, as their variation is very small. In contrast, it is essential for memristive devices as they may introduce errors. This variation is mentioned as a percentage in case reported; the percentage represents (HLR-LLR)/μ_{LRS}. Note that the articles not always describe the variation in this form. For example, Jain *et al.* assume that the MTJ oxide thickness has 2% variation. In such cases, we converted the given variation type into a resistance variation. The only exception is Chen *et al.*, as they verified their schemes using real RRAM devices.

The last two columns show the worst case latency in number of clock cycles required for a logic operation and whether multiple operands are supported simultaneously, respectively. Some articles claim that they support multiple operands (executed by activating multiple rows in the memory), at least for the OR operation [5], [11], [25]. However, the ability of the sense amplifier to be able to distinguish between the outcomes is extremely difficult or even not possible.

All the bulk bitwise operations based on memristive devices that are listed in Table II are implemented with similar hardware structures. Fig. 2a represents such a hardware structure. The bulk bitwise operations are special read operations. Rows are activated by drivers (represented by triangles) and operate

TABLE II
Bulk Bitwise Boolean Logic Operations

Reference	Scheme name	Technology	Supported operations	Variation considered?	Longest delay	Multirow
[21]	Seshadri *et al.*	DRAM	AND, OR	No	4	No
[22]	Ambit	DRAM	AND, OR, NOT	No	4	No
[23]	DRISA	DRAM	NOR	No	2	No
[24]	Jeloka *et al.*	SRAM	AND, NOR	No	1	Yes
[5]	Pinatubo	NVM	AND, OR, XOR, NOT	No	2	Yes
[11]	Scouting logic	Memristor	AND, OR, XOR	Yes, $< 10\%$	1	Yes
[25]	MPIM	Memristor	AND, OR, XOR	Yes, 10%	1	Yes
[26]	IMP/NMP	STT-MRAM	AND, OR, NOT	No	1	No
[27]	HielM	STT-MRAM	AND, OR, XOR	Yes, $< 5\%$	4	No
[28]	STT-CiM	STT-MRAM	AND, OR, NOT	Yes, $< 10\%$,	1	No
[29]	IMCS2	STT-MRAM	AND, OR, XOR, NOT	Yes, $< 5\%$	1	No
[30]	Chen *et al.*	RRAM	AND, OR, XOR	Yes, measurement	1	No
This work	ESL	RRAM	AND, OR, NOT	Yes, $> 100\%$	1	No



IN_i	1	1	0	0
IN_j	1	0	1	0
OR		1		0
AND	1		0	

(a) Hardware structure

(b) Parallel resistance

Fig. 2. Working principle of memristive bulk bitwise operations.

at read voltage V_r. The memory cells consist of a 1T1R structure, i.e. a transistor in parallel with a memristive device (represented by a rectangle) such as RRAM or STT-MRAM. During bulk bitwise operations, two or more rows are selected at the same time. The total current through the two cells in each column is compared with a reference current I_{ref} by the sense amplifier. The output of the sense amplifier is the desired result of the required Boolean function. There are also other forms of comparison such as using a voltage reference. However, the essence is to compare the equivalent resistance of two parallel cells with a reference resistance.

The resistance distribution of two cells (IN$_i$ and IN$_j$) that are connected in parallel is shown in the top of Fig. 2b. Here, we assume that a logic one and zero are represented by a low and high resistance, respectively. There are three possible combinations: 1) both cells in HRS (2H), 2) one cell in LRS and one in HRS (1L1H), and 3) both cells in LRS (2L). To conduct an OR operation, we need to set a reference R_O to distinguish between the cases 2H and 1L1H, as indicated by the truth table in Fig. 2b. If the device has a large margin between LRS and HRS like the ones listed in Table I, it would be easy to define R_O. However, it is difficult or impossible to set a reference R_A to distinguish 1L1H from 2L in case an AND operation is implemented. When HLR/LLR$>$ 2, which is the case of many devices in Table I, the 2L and 1L1H regions overlap. Thus, a correct AND operation cannot be guaranteed regardless of the value of R_A.

Chen *et al.* proposed a self-write termination (SWT) scheme for RRAM devices to avoid the above mentioned problem [30]. The scheme adds a loop-back from the cell to the driver during write operations. When the programmed memristive device reaches the desired resistance, the writing process is terminated. SWT's main target is to reduce the write energy and the standard deviation of the resistance distribution. However, SWT schemes also have challenges such as premature ending of write operations that may lead to unstable resistance states [32]. In addition, it is also difficult to achieve HLR/LLR$<$ 2 using SWT schemes for some devices [32].

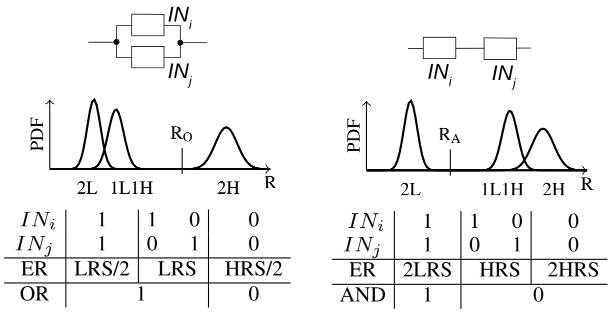
IV. Enhanced Scouting Logic

In this section, we first introduce the general concept behind ESL. Thereafter, we present the implementation details.

A. *General Concept*

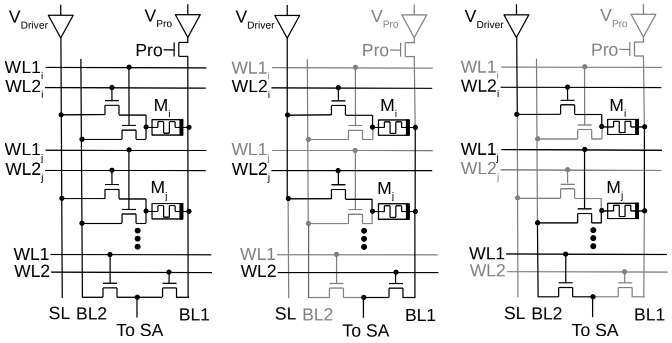
To prevent the possibility of overlapping regions of different equivalent resistance values, ESL changes the connection of memristive devices based on the operation type. Despite the large variations, a reference resistance can be still safely selected. As shown in Fig. 2b, there are no issues in selecting a reference resistance R_O even if the regions 2L and 1L1H are overlapping. Hence, this part of the circuit does not have to be changed. This is shown in Fig. 3a. However, for AND operations, ESL changes the circuit structure and connects the input cells in series. The overall resistance's distribution will be similar to the one shown in Fig. 3b. Notably, the equivalent resistance (ER) of 1L1H would be approximately HRS instead of LRS. Therefore, there is a relatively large margin between 2L and 1L1H, and the reference R_A can be easily selected.

Considering the fact that most devices have a large margin, only one reference is needed in ESL. From Fig. 3, we observe that the following relations must satisfy: LRS$<R_O<$HRS$/2$ (as ER of 1L1H$\approx$LRS and 2H$\approx$HRS$/2$ for the OR operation) and 2LRS$<R_A<$HRS (as ER of 2L=2LRS and 1L1H$\approx$HRS for the AND operation). From these relations we can easily observe that for given $R_O = R_A = R_{ref}$ the equation 2LRS$<R_{ref}<$HRS$/2$ must hold. Hence, a single reference can be used when the resistance margin is larger than $4x$. Note



(a) OR operation (b) AND operation

Fig. 3. Resistance distribution of Enhanced Scouting Logic.



(a) Circuit (b) OR operation (c) AND operation

Fig. 4. ESL circuit.

that all the devices listed in Table I meet this requirement. This feature makes the reference circuit much simpler as compared to those needed in previous bulk bitwise operation schemes [5], [11], [25].

B. Implementation

The circuit to implement ESL is shown in Fig. 4a. For simplicity reasons, Fig. 4a illustrates the concept using only two cells (M_i and M_j) located in a single column. Other cells in this column and other columns have similar structures. Besides the standard bitline BL1 in a 1T1R array, a second bitline BL2 is added to each column to be able to connect two devices in series which is needed for the AND operation. The two bitlines share one sense amplifier using two transistors (controlled by voltages WL1 and WL2, respectively). One more transistor is added to each cell to connect the device to BL2.

The values of the control signals during each operation are shown in Table III. The operations are the normal write/read and the bitwise OR/AND operations. The table contains for each of the control lines in Fig. 4 the voltage values per operation type. When we write M_i (i.e. perform a SET or RESET operation), Pro and $WL2_i$ are enabled. The voltage level of the Driver and V_{Pro} depend on the value to be programmed. For instance, the voltage applied to M_i is V_{SET} if we want to program a logical one. When we read cell M_i, $WL2_i$ and WL2 are enabled and Driver's voltage changes to V_{Read}. In our implementation, V_{Read}=V_{dd}. However, it can also be a different value, e.g., to prevent a drift of the memristor state. During OR/AND operations, the Driver's voltage is also V_{Read}. Pro is disabled to disconnect the programming driver to the circuit. In an OR operation between M_i and M_j, $WL2_i$ and $WL2_j$ are on. As a result, $Driver_i$ and $Driver_j$ drive the two cells in parallel. Fig. 4b illustrates these signals by coloring disabled transistors and unused wires gray. In an AND operation, however, only one driver (e.g., $Driver_i$) is connected to the cells as shown in Fig. 4c. $WL1_j$ and WL1 are enabled to connect the two cells to the SA in series.

TABLE III
CONTROL SIGNALS FOR ESL IMPLEMENTATION

	Write	Read	OR	AND
V_{Driver}	0/V_{RESET}	V_{Read}	V_{Read}	V_{Read}
V_{Pro}	V_{SET}/0	–	–	–
Pro	V_{dd}	0	0	0
$WL1_i$	0	0	0	0
$WL2_i$	V_{dd}	V_{dd}	V_{dd}	V_{dd}
$WL1_j$	0	0	0	V_{dd}
$WL2_j$	0	0	V_{dd}	0
WL1	0	0	0	V_{dd}
WL2	0	V_{dd}	V_{dd}	0

We use the state-of-the-art CSB-SA [33] sense amplifier in our design. It operates in three phases. In the first phase, it captures the inputs, i.e., the currents through the memristor cell and the reference resistor are sampled and stored in two capacitors, respectively. In the second phase, the stored charges on the capacitor are amplified. Finally, the output latch is enabled during the third phase and a digital output is generated. Note that other sense amplifiers could also be used. The selected sense amplifier mainly depends on the trade-off between performance and area.

V. SIMULATION RESULTS

In this section, we first present the simulation setup. Subsequently, we validate ESL using SPICE simulation. Finally, we conduct Monte Carlo simulation with ESL and other logic schemes and evaluate their results.

A. Simulation Setup

To verify the proposed scheme, we use SPICE simulations. The circuit in Fig. 4 is implemented and connected to the CBS-SA sense amplifier. The simulation parameters are summarized in Table IV. The supply voltage V_{dd} is set to $0.9\,\mathrm{V}$. The resistive device of the 2T1R cell is implemented using the ASU model [34] configured with the RRAM parameters presented in [20]. The RRAM device is implemented with $28\,\mathrm{nm}$ technology. To match the transistor sizes, we therefore adopt the $32\,\mathrm{nm}$ PTM model [35], which is the closest technology node for CMOS transistors in PTM. The sense amplifier uses a reference resistor that is set to $160\,\mathrm{k\Omega}$, which is the geometric average of HLR and LHR of the RRAM [20]. The three phases of the sense amplifier are set to be $1\,\mathrm{ns}$.

To verify the robustness of ESL against resistance variations, all the corner cases have been simulated. These corner

TABLE IV
SPICE SIMULATION PARAMETERS

Parameter		Description	Value
V_{dd}		CMOS power supply	0.9 V
LLR		Low low resistance	10 kΩ
HLR		High low resistance	50 kΩ
LHR		Low high resistance	500 kΩ
HHR		High high resistance	500 MΩ
F		Technology node	28 nm
R_{ref}		Reference resistance	160 kΩ
t_s		Sense amplifier phase	1 ns
Monte Carlo simulation			
N_σ		Number of sigma	3
N		Number of iterations	10,000
R_L	μ	Mean resistance	30 kΩ
	σ	Standard deviation	0.5
R_H	μ	Mean resistance	16.6 MΩ
	σ	Standard deviation	1.68

cases are derived from the extreme resistance values for each of the two inputs. The four extreme resistance values are LLR, HLR, LHR, and HHR. Therefore, there are 16 combinations for two inputs. All of these combinations are simulated for both AND and OR operations, respectively.

To compare our scheme with the with the state-of-the-art schemes, we also conducted Monte Carlo simulations of the AND gates of Scouting Logic [11], Pinatubo [5], and ESL, respectively. We implemented these circuits according to their papers. Subsequently, we configured the parameters of two lognormal distributions as listed in the bottom part of Table IV. Finally, we performed 10,000 Monte Carlo simulations for each design scheme.

B. Validation of ESL

Based on the simulation of the 16 corner cases, we conclude that the proposed ESL implementation is robust against resistance variations for both the AND and OR operations. Fig. 5 shows the waveform of all the 16 test cases for the AND operation. The x axis represents time, and the y axis indicates the voltage. The bitline voltages in the 16 test cases are divided into three groups according to the logic values of the input cells. The 2H group is colored red, 1L1H cyan, and 2L orange. The voltage of the reference bitline is colored black. The three phases can be easily identified in the figure. The resistance of the input cells is sampled in the first phase. The higher the voltage in the graph, the higher the resistance. We refer the readers to [33] for more details about the working principle of the sense amplifier. The reference is set between 1L1H and 2L as shown in Fig. 3b. Note that the 2H and 1L1H groups overlap. For example, HLR+HHR> 2LHR; while HLR+HHR belongs to 1L1H and 2LHR to 2H, the resistance of 1L1H can be still higher than 2H. After the two-step amplification in phases 2 and 3, the 2L cases generate a logical one while the remaining cases a logical zero. These results are consistent with the AND's truth table.

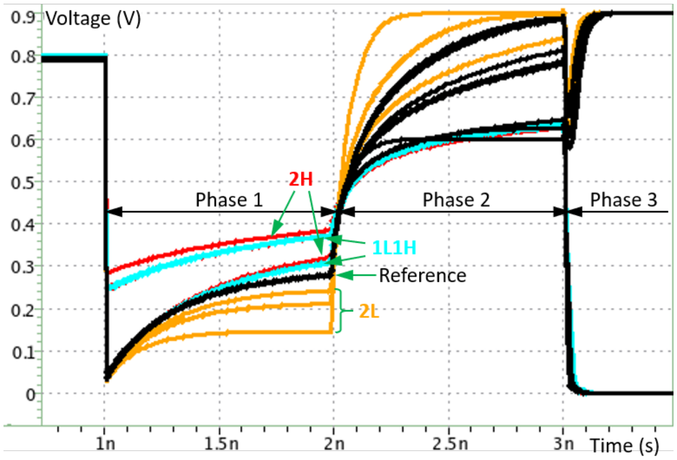


Fig. 5. SPICE simulation results of AND operation.

TABLE V
THE NUMBERS OF FAILED CASES IN 10,000 MONTE CARLO ITERATIONS

	HH	HL	LH	LL	Total
Scouting Logic	0	75	97	202	374
Pinatubo	0	142	176	332	650
ESL (This work)	0	0	0	0	0

The waveform of the OR operation is similar to Fig. 5 and hence not included in the paper for simplicity. Also the 16 cases verified the correctness of ESL.

C. Comparison with State of the Art

Table V summarizes for Scouting Logic, Pinatubo, and ESL the number of failures observed during Monte Carlo simulations. A simulation is considered to fail when a wrong logic value is observed (an output voltage lower than 40% of V_{dd} is considered a logic zero and higher than 60% of V_{dd} a logic one) or when the sense amplifier output ranges between 40% and 60% of the V_{dd} value. Each row in the table represents a logic scheme. The columns are the four resistance combination of the two input memristors, i.e. HH, HL, LH, and LL, and the numbers represent the total failed cases. The table clearly shows that ESL is robust against variations, while the other schemes can practically not be used. Note that we configured the reference resistance for Pinatubo and Scouting Logic with different values and repeated the simulations. Only the simulation groups that had the smallest total number of failed cases are reported in Table V. It is impossible to find a perfect value that avoids failures for these designs, as the input resistance ranges of 1L1H and 2L overlap.

D. Discussion

The main advantage of ESL is that it is able to deal with large resistance variations. Many previous works overlooked this fact and did not consider this during design. ESL has besides this benefit also other benefits. It requires only one reference resistor which makes it simpler than previous works.

Finally, like other bulk bitwise logic schemes, ESL does not write to memristors during process; hence, it does not affect the endurance during logic operations.

To enhance the robustness, we designed a 2T1R memory cell structure, which is different from the typical cases such as 1R, 1T1R, or 1S1R. Compared with these structures, 2T1R requires more control signals and has a larger cell area.

VI. Conclusion

Today's RRAM devices suffer from large resistance variations. Existing Boolean logic schemes that based on resistance sensing cannot guarantee the correctness of these operations under such variations. To enhance the robustness, we have proposed a new scheme that senses the resistance via two paths for AND and OR operations, respectively. SPICE simulation and Monte Carlo analysis have proved the robustness of our scheme. This scheme brings the memristive logic circuits a step closer to reality as the robustness is a primary concern.

Acknowledgment

This work was supported by the European Unions Horizon 2020 Research and Innovation Program through the project MNEMOSENE under Grant No. 780215.

References

[1] S. Hamdioui, S. Kvatinsky, G. Cauwenberghs, L. Xie, N. Wald, S. Joshi, H. M. Elsayed, H. Corporaal, and K. Bertels, "Memristor for computing: Myth or reality?" in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017, pp. 722–731.

[2] J. Hennessy and D. Patterson, *Computer Architecture: A Quantitative Approach*, 6th ed. Amsterdam, Netherlands: Elsevier, Nov. 2017.

[3] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb 2014, pp. 10–14.

[4] S. Hamdioui, L. Xie, H. A. D. Nguyen, M. Taouil, K. Bertels, H. Corporaal, H. Jiao, F. Catthoor, D. Wouters, L. Eike, and J. van Lunteren, "Memristor based computation-in-memory architecture for data-intensive applications," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, ser. DATE '15. San Jose, CA, USA: EDA Consortium, 2015, pp. 1718–1725.

[5] S. Li, C. Xu, Q. Zou, J. Zhao, Y. Lu, and Y. Xie, "Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories," in *Proceedings of the 53rd Annual Design Automation Conference*. New York, NY, USA: ACM, 2016, pp. 173:1–173:6.

[6] J. Yu, H. A. Du Nguyen, L. Xie, M. Taouil, and S. Hamdioui, "Memristive devices for computation-in-memory," in *2018 Design, Automation Test in Europe Conference Exhibition*. IEEE, March 2018, pp. 1646–1651.

[7] S. Yu and P. Y. Chen, "Emerging memory technologies: Recent trends and prospects," *IEEE Solid-State Circuits Magazine*, vol. 8, no. 2, pp. 43–56, Spring 2016.

[8] A. Chen, "A review of emerging non-volatile memory (NVM) technologies and applications," *Solid-State Electronics*, vol. 125, no. Supplement C, pp. 25 – 38, 2016.

[9] G. Snider, "Computing with hysteretic resistor crossbars," *Applied Physics A*, vol. 80, no. 6, pp. 1165–1172, 2005.

[10] J. Borghetti, G. Snider, P. Kuekes, J. Yang, D. Stewart, and R. Williams, "'Memristive' switches enable 'stateful' logic operations via material implication," *Nature*, vol. 464, no. 7290, pp. 873–876, 2010.

[11] L. Xie, H. A. D. Nguyen, J. Yu, A. Kaichouhi, M. Taouil, M. Al-Failakawi, and S. Hamdioui, "Scouting logic: A novel memristor-based logic design for resistive computing," in *2017 IEEE Computer Society Annual Symposium on VLSI*, July 2017, pp. 176–181.

[12] L. Zhang, S. Cosemans, D. J. Wouters, G. Groeseneken, M. Jurczak, and B. Govoreanu, "Cell variability impact on the one-selector one-resistor cross-point array performance," *IEEE Transactions on Electron Devices*, vol. 62, no. 11, pp. 3490–3497, Nov 2015.

[13] A. Fantini, L. Goux, R. Degraeve, D. J. Wouters, N. Raghavan, G. Kar *et al.*, "Intrinsic switching variability in HfO2 RRAM," in *5th IEEE International Memory Workshop*, May 2013, pp. 30–33.

[14] A. Grossi, E. Nowak, C. Zambelli, C. Pellissier, S. Bernasconi, G. Cibrario *et al.*, "Fundamental variability limits of filament-based RRAM," in *2016 IEEE International Electron Devices Meeting*, Dec 2016, pp. 4.7.1–4.7.4.

[15] R. Han, P. Huang, Y. Zhao, Z. Chen, L. Liu, X. Liu *et al.*, "Demonstration of logic operations in high-performance RRAM crossbar array fabricated by atomic layer deposition technique," *Nanoscale Research Letters*, vol. 12, no. 1, p. 37, Jan 2017.

[16] X. Hong, P. A. Dananjaya, S. Krishnia, W. Gan, D. J. Loy, F. Tan *et al.*, "A novel geometry of ECM-based RRAM with improved variability," *Journal of Physics D: Applied Physics*, 2018.

[17] Y. Fang, Z. Yu, Z. Wang, T. Zhang, Y. Yang, Y. Cai, and R. Huang, "Improvement of HfOx-Based RRAM device variation by inserting ALD TiN buffer layer," *IEEE Electron Device Letters*, vol. 39, no. 6, pp. 819–822, June 2018.

[18] A. Mehonic, M. Munde, W. Ng, M. Buckwell, L. Montesi, M. Bosman *et al.*, "Intrinsic resistance switching in amorphous silicon oxide for high performance siox reram devices," *Microelectronic Engineering*, vol. 178, pp. 98 – 103, 2017, iNFOS.

[19] A. Bricalli, E. Ambrosi, M. Laudato, M. Maestro, R. Rodriguez, and D. Ielmini, "Siox-based resistive switching memory (RRAM) for crossbar storage/select elements with high on/off ratio," in *2016 IEEE International Electron Devices Meeting*, Dec 2016, pp. 4.3.1–4.3.4.

[20] H. Lv, X. Xu, P. Yuan, D. Dong, T. Gong, J. Liu *et al.*, "Beol based rram with one extra-mask for low cost, highly reliable embedded application in 28 nm node and beyond," in *2017 IEEE International Electron Devices Meeting*, Dec 2017, pp. 2.4.1–2.4.4.

[21] V. Seshadri, K. Hsieh, A. Boroum, D. Lee, M. A. Kozuch, O. Mutlu *et al.*, "Fast bulk bitwise AND and OR in DRAM," *IEEE Computer Architecture Letters*, vol. 14, no. 2, pp. 127–131, July 2015.

[22] V. Seshadri, D. Lee, T. Mullins, H. Hassan, A. Boroumand, J. Kim *et al.*, "Ambit: In-memory accelerator for bulk bitwise operations using commodity DRAM technology," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*. New York, NY, USA: ACM, 2017, pp. 273–287.

[23] S. Li, D. Niu, K. T. Malladi, H. Zheng, B. Brennan, and Y. Xie, "Drisa: A DRAM-based reconfigurable in-situ accelerator," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*. New York, NY, USA: ACM, 2017, pp. 288–301.

[24] S. Jeloka, N. B. Akesh, D. Sylvester, and D. Blaauw, "A 28 nm configurable memory (TCAM/BCAM/SRAM) using push-rule 6T bit cell enabling logic-in-memory," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 4, pp. 1009–1021, April 2016.

[25] M. Imani, Y. Kim, and T. Rosing, "Mpim: Multi-purpose in-memory processing using configurable resistive memory," in *22nd Asia and South Pacific Design Automation Conference*, Jan 2017, pp. 757–763.

[26] W. Kang, H. Wang, Z. Wang, Y. Zhang, and W. Zhao, "In-memory processing paradigm for bitwise logic operations in STT-MRAM," *IEEE Transactions on Magnetics*, vol. 53, no. 11, pp. 1–4, Nov 2017.

[27] F. Parveen, Z. He, S. Angizi, and D. Fan, "Hielm: Highly flexible in-memory computing using STT MRAM," in *23rd Asia and South Pacific Design Automation Conference*, Jan 2018, pp. 361–366.

[28] S. Jain, A. Ranjan, K. Roy, and A. Raghunathan, "Computing in memory with spin-transfer torque magnetic RAM," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 3, pp. 470–483, March 2018.

[29] F. Parveen, S. Angizi, Z. He, and D. Fan, "Imcs2: Novel device-to-architecture co-design for low-power in-memory computing platform using coterminous spin switch," *IEEE Transactions on Magnetics*, pp. 1–14, 2018.

[30] W. H. Chen, W. J. Lin, L. Y. Lai, S. Li, C. H. Hsu, H. T. Lin *et al.*, "A 16mb dual-mode ReRAM macro with sub-14ns computing-in-memory and memory functions enabled by self-write termination scheme," in *2017 IEEE International Electron Devices Meeting*, Dec 2017, pp. 28.2.1–28.2.4.

[31] D. Knuth, *The Art of Computer Programming: Bitwise Tricks & Techniques*, 1st ed. Boston, MA: Addison-Wesley, Mar. 2009.

[32] Z. Wang, Y. Liu, A. Lee, F. Su, C. P. Lo, Z. Yuan *et al.*, “A 65-nm reram-enabled nonvolatile processor with time-space domain adaption and self-write-termination achieving $> 4\times$ faster clock frequency and $> 6\times$ higher restore speed,” *IEEE Journal of Solid-State Circuits*, vol. 52, no. 10, pp. 2769–2785, Oct 2017.

[33] M. F. Chang, S. J. Shen, C. C. Liu, C. W. Wu, Y. F. Lin, Y. C. King *et al.*, “An offset-tolerant fast-random-read current-sampling-based sense amplifier for small-cell-current nonvolatile memory,” *IEEE Journal of Solid-State Circuits*, vol. 48, no. 3, pp. 864–877, March 2013.

[34] P. Y. Chen and S. Yu, “Compact modeling of rram devices and its applications in 1T1R and 1S1R array design,” *IEEE Transactions on Electron Devices*, vol. 62, no. 12, pp. 4022–4028, Dec 2015.

[35] NIMO Group, ASU, “Predictive technology model.” [Online]. Available: http://ptm.asu.edu/