Sub-pJ Consumption and Short Latency Time in RRAM Arrays for High Endurance Applications

Gilbert Sassine, Cécile Nail, Luc Tillie, Diego Alfaro Robayo, Alexandre Levisse, Carlo Cagli, Khalil El. Hajjam, Jean-François Nodin, Elisa Vianello, Mathieu Bernard, Gabriel Molas, Etienne Nowak
CEA-LETI, MINATEC Campus
Grenoble, France
phone: (+33) 438 781 066, gilbert.sassine@cea.fr

***Abstract*—In this paper, programming operations are optimized for low energy consumption and short latency time applications in RRAM kb arrays. Origin of consumption (role of pulse's time, programming current and voltage in SET and RESET operations) is quantified on HfO_2 oxide based RRAM technology. Specific patterns are evaluated to reduce latency time and energy consumption in memory devices. Innovative circuit with *on the fly* switching detection is proposed, allowing to reduce programming consumption down to single pJ operation in large memory arrays.**

***Index Terms*—Energy consumption, latency time, RRAM, smart programming**

I. Introduction

Resistive random access memories RRAM are expected to play a major role in future semiconductor industry due to their promising features such as fast switching speed, high retention, good endurance, and great compatibility with CMOS technology [1]. Energy consumption becomes a critical concern, in data center, due to popularity of cloud storage and mobile devices using 10 to 40 Watthours per device a day [2]. It is thus urgent to develop lower energy consuming devices. Among them, memory devices significantly contribute to the overall energy consumption. At single cell level, RRAM can theoretically be operated at 10pJ [1]. However, looking at statistics, larger programming time can be required to insure sufficient window margin (WM) for a high number of cycles, due to intrinsic RRAM variability [3]. Programming energy can drastically increases in high density and high endurance applications [4]. In this paper, we investigate how technology, optimized programming schemes and design can help reducing RRAM consumption, providing good endurance and short latency, based on experiments performed on 1T1R kb arrays.

II. Results and Discussion

A. Context

Electrical characterization was performed on TiN / HfO_2 5nm / PVD Ti 5nm RRAM kb arrays integrated on 130nm CMOS logic on top of Metal 4 in 1T1R configuration. Fig. 1.a shows endurance characteristics measured on kb arrays with 60µA programming current. While more than one decade separates the median values of low resistive state (LRS) and high resistive state (HRS), overlap appears on the tails, reducing active memory window margin (WM). Resistance distributions in normal scale are presented in Fig.1.b. 4σ can hardly be achieved with good ON and OFF states distinction, particularly after cycling where we can clearly see that LRS and HRS overlap after 10^7 cycles. Fig.2.a plots WM at 2σ extracted from 10^5 cycles versus energy consumption. Various SET and RESET voltages, times (200ns to 100µs) and currents (10µA to 100µA) were used. Consumption was estimated at the 1st order by $E_c = V_{prog} x I_{prog} x T_{prog}$. We observe that large WM required to target dense memory arrays can hardly be reached under 10pJ. In order to assure sufficient WM, minimum I_{prog} of 50µA and T_{prog} of ~200ns are identified. In conclusions, in memory arrays, reducing time is more efficient than reducing programming current to limit energy consumption and ensure sufficient WM.

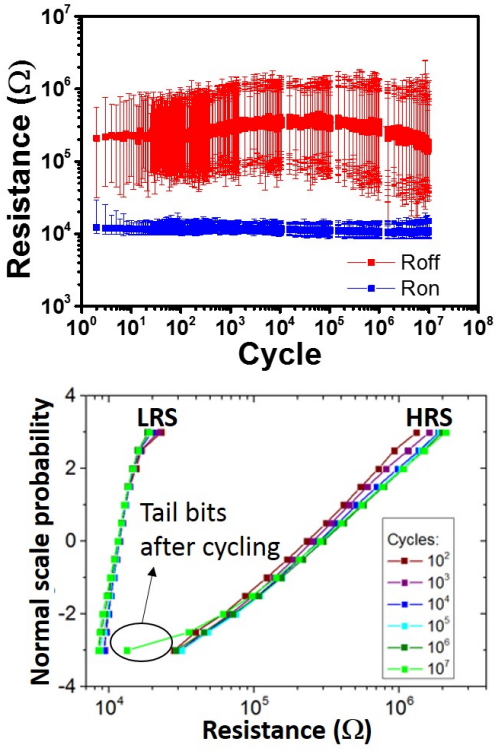

Figure 1 (a) RRAM endurance measured on 4kb array for I_{prog} =60uA (2 σ dispersion is showed). (b) Typical LRS and HRS resistance distributions for different number of cycles. Tail bits close window margin after cycling.

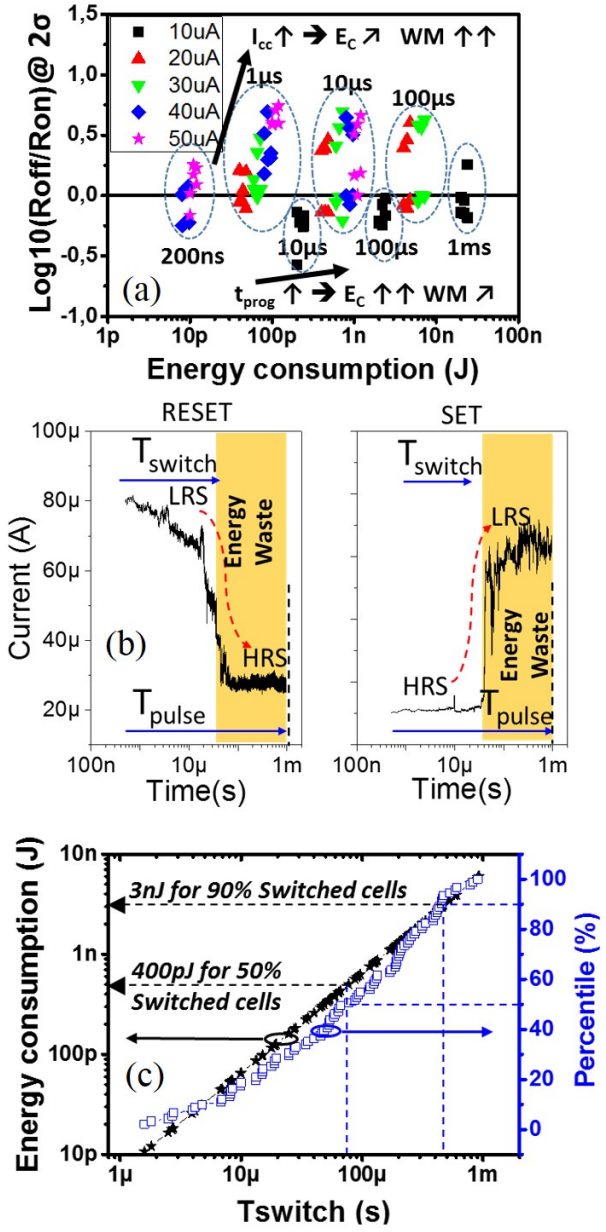

Figure 2 (a) Window margin (worst case) extracted at 10^5 cycle from endurance characteristics versus energy consumption for various programming time and current. (b) Current versus time characteristics during CVS measurements for SET and RESET. Difference between pulse time and switching time is responsible for energy consumption in SET mode. In RESET, consumption results from T_{switch}. (c) SET energy consumption distribution as a function of RRAM switching time. Wide T_{switch} distribution leads to slow bits and thus high energy consumption.

Fig.2.b illustrates the origin of energy consumption during SET and RESET. Difference between pulse length T_{pulse} and RRAM switching time T_{switch} is responsible for SET consumption while in RESET, consumption results from T_{switch}. However, large switching time intrinsic dispersion makes short pulse programming difficult, and is responsible for high energy consumption especially for high density memories (Fig.2.c) for which long pulses are required to program large memory arrays.

B. *Programming schemes*

In the aim of improving energy consumption different programing schemes were evaluated. First, constant voltage (CVS) is applied to our RRAM [5]. Switching time is measured as function of the applied voltage (Fig.3.a), showing a wide intrinsic distribution. Up to 5 decades spread can be seen for different applied voltages. Slow bits impose large programming time, while most of the memory cells can be operated with a much lower time and thus lower energy. CVS is thus not adapted for low programming energy. In a second approach, ramp voltage stress (RVS) is applied [5]. Fig.3.b shows that switching voltage increases with RVS ramp speed in SET and RESET. RVS characteristics can be translated to switching time versus switching voltage curves using an analytical model [6-7] (Fig.3c-e).

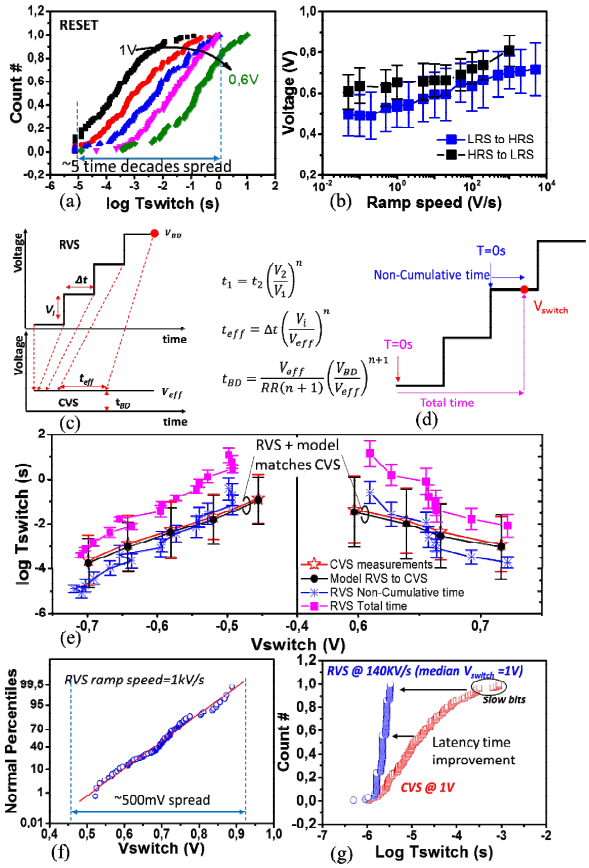

Figure 3 (a) RESET switching time distributions for different applied voltages extracted from CVS measurements showing wide intrinsic dispersion. (b) Switching voltage versus ramp speed extracted from RVS measurements. (c) Schematics of RVS signal and corresponding Voltage stress for a constant applied bias (CVS equivalence) and an analytical model allowing to translate RVS characteristics to T_{switch} (V_{switch}) curves. (d) Schematic of non-cumulative time versus total time for time extraction in RVS. (e) T_{switch} versus V_{switch} characteristics extracted from CVS (red) and RVS (blue and magenta) measurements. Non-cumulative RVS (blue) is extracted by considering the impact of the last applied pattern with the highest voltage while cumulative time (magenta) is extracted by considering the time effect of the whole pattern. Black line is the converted RVS voltage to CVS time using an analytical model. (f) Typical switching voltage distribution from RVS measurements. (g) Latency time distribution for RVS and CVS measurements.

The model shows that the cell is mostly sensitive to the last applied pattern, with the highest applied voltage (not shown here). Indeed, for a given switching voltage, switching time in CVS mode is close to the equivalent stress time applied at V_{switch} in RVS (RVS non-cumulative time in Fig.3.e). For a given ramp speed, switching voltage is spread over ~500mV (Fig.3.f). Thus, voltage increase in RVS gradually increases programming strength, reducing impact of intrinsic RRAM switching time dispersion (Fig.3.a).Programming time is then strongly reduced in comparison with CVS (Fig.3.g), reducing energy consumption. On the other hand, and since in RVS the switching is sensitive to the last applied patterns, the device undergoes less time stress, potentially leading to higher reliability.

C. *On the fly smart programming consumption*

Based on the results above, a sequence of 50ns pulses in RVS or Pulsed (PVS) mode are applied (Fig.4.a). Pulse train is stopped when a current variation is detected during the pulse (*on the fly* smart programming). Voltage stress in PVS can be adjusted to limit the number of applied pulses, while maximum RVS voltage is identified to switch 100% of RRAM cells (Fig.4b-c).

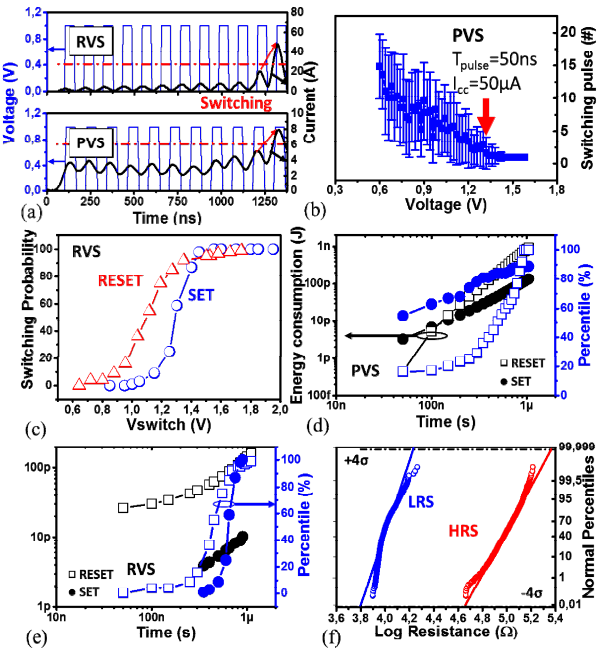

Figure 4 (a) Applied WL pattern in RVS and PVS modes (SET operation), and measured RRAM cell current. Wordline is pulsed (50ns) while Bitline voltage is quasistatic: constant in PVS and ramped in RVS. Cell switching is detected by measuring the current during the stress (on the fly smart programming). (b) Number of pulses required in PVS to switch the RRAM versus pulse amplitude for pulse time of 50ns. ~1.2V applied voltage is identified to keep a moderate number of applied pulses to SET the RRAM. (c) SET and RESET switching voltages in RVS. All the cells could be switched at the end of the ramp. (d) PVS and (e) RVS SET and RESET consumptions (black symbols) as function of latency time. To switch 100% of the cells, consumption gradually increases. (f) Example of LRS and HRS resistance distributions for PVS with I_{CC}=100µA. Smart procedure allows to keep a window margin even at 4σ.

Fig.4.d-e show PVS and RVS SET and RESET consumptions as function of latency time. Smart RVS (pulse ramp is stopped when the cell switches) offers lower consumption (100pJ in RESET and 10pJ in SET) to program all the RRAM array in comparison to smart PVS. Smart procedure allows to improve window margin up to >4σ (Fig.4.f).

D. *New design for on the fly smart programming*

A specific circuit was developed in order to stop the applied voltage as soon as a resistance variation of the memory cell occurs (Fig.5). The proposed circuit concept consists in three coupled blocks: *(i)* a local signal generator (green), *(ii)* a bias generator (red) and *(iii)* a write circuit (blue). When RRAM is programmed with a standard current mirror LowDropOut (LDO) association [8], Vd node voltage falls (SET) or rises (RESET). Consequently, Prog_ack signal switches and is used in the local signal generator. Local Enable Signal (L_EN) is generated from the Write Enable signal (Wr_EN) and Prog_ack. L_EN is precharged to vdd when Wr_EN switches to vdd and is then pulled down to gnd when Prog_ack switches, whatever Wr_EN signal state. L_EN is used to stop programming operation in order to reduce power consumption. L_CK signal is generated by an AND operation between L_EN and the global clock signal (CK). L_EN and L_CK are then used to control the Bias Generator Block. In this block, FFi is initially set to '1' while FF0-FF(n-1) are reset to '0'. Then, at each L_CK cycle, the '1' is shifted in flip-flops from FF0 to FF(n-1). This generates the address signal A<0:n-1> controlling the multiplexer. This latter consists in transmission gates connecting the voltages V(0:n-1) to V_{prog} depending on A<0:n-1> code. Thereby, on time, V_{prog} is biased sequentially from V_0 up to V_{n-1} and rises at each L_CK cycle. On the other hand, when L_EN='0', V_{prog} is pulled down to gnd. Finally, V_{prog} signal controls the LDO which bias the array line voltage.

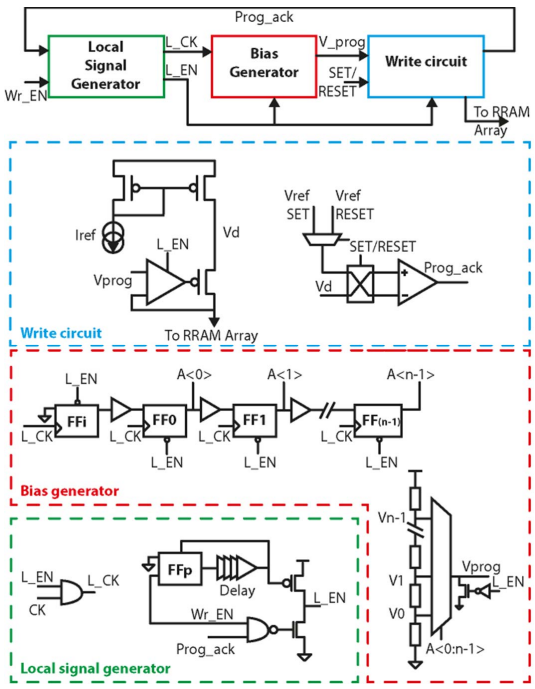

Figure 5 Schematics circuit allowing on the fly smart programming: SET or RESET pulse is stopped as soon as the RRAM switching occurs.

Fig.6 shows transient operation of previously presented circuit during a SET operation simulated with a 130nm thick oxide transistors CMOS technology. Fig.6.a presents the full pattern generated for three different RRAM devices corners (#1: Fast, #2: Typical, #3: Slow). For each RRAM switch V_{prog} voltage automatically increases to adapt to the RRAM optimal programming voltage. Then, Prog_ack signal switches with the RRAM, stopping the programming operation. L_CK signal is also shown compared to the global clock signal (red). Fig.6.b presents a zoom in the programming switch showing current pulse in the RRAM, programming voltage and Prog_ack signal switch. For the considered CMOS technology, a 3ns write termination delay is achievable.

Thanks to this, RRAM corresponding consumption is calculated. Energy distributions are presented in figure 7.a. Different ramp speeds (total pattern time) are applied. As we can clearly see, energy consumption is significantly reduced by increasing ramp speed (Fig.7.a). Then, extrapolations were made (magenta curve in fig.7.a) assuming aggressive plateaus durations down to 50ns (requiring RF environment). In the case of 4σ statistics, SET and RESET programming energies respectively reach 1pJ and 10pJ (mean values 500fJ and 1pJ), which is more than two order of magnitude lower to non-smart programming algorithms (Fig.7.b), and 2 decades lower than CVS operation (Fig.7.c).

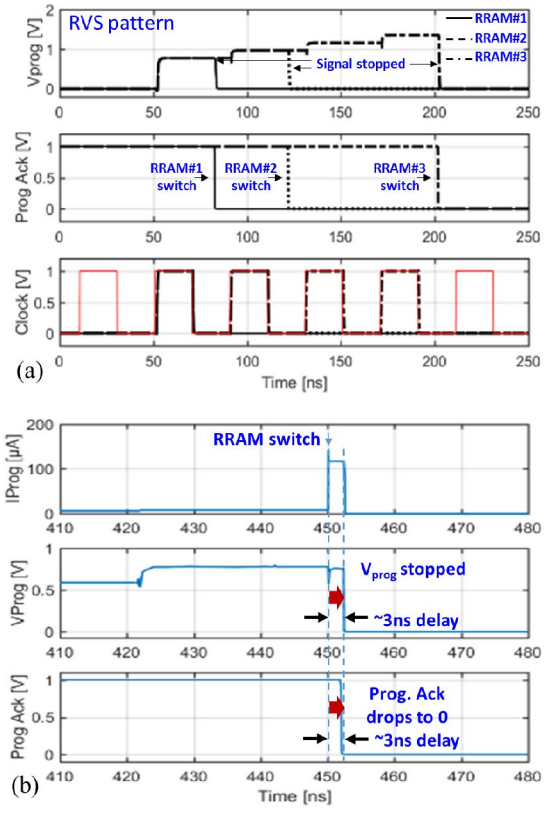

Figure 6 (a) Applied RVS patterns in the circuit described in Fig.5. 3 RRAM switching are represented, leading to RVS signal stop (Prog. Ack drops to 0). (b) I_{prog} and V_{prog} for a specific RRAM in the circuit; efficient smart operation is achieved: stress voltage stops ~3ns after RRAM switch occurs.

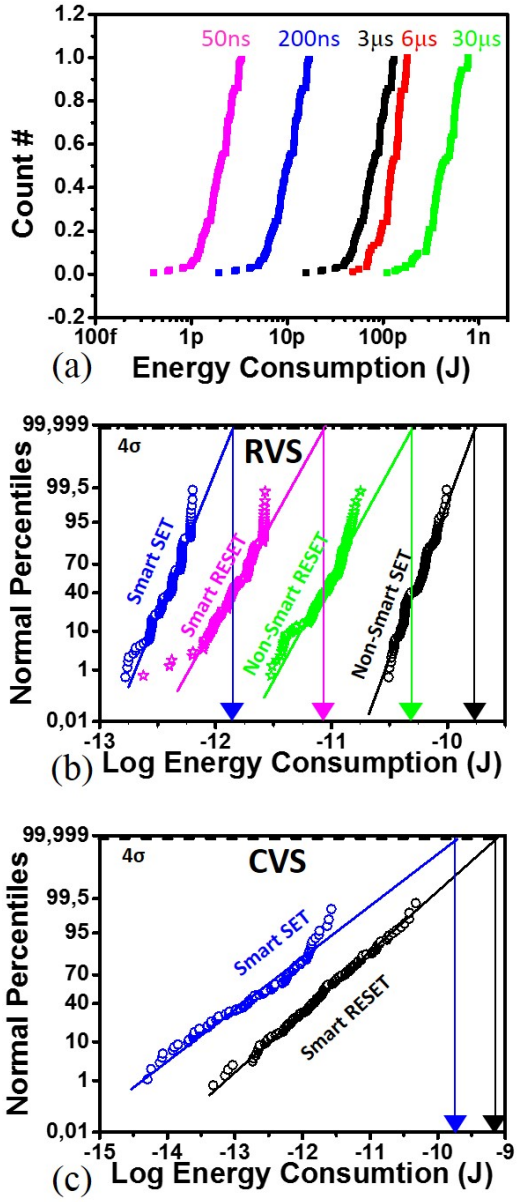

Figure 7 (a) Energy consumption distribution (SET+RESET) in RVS using smart circuit (Fig.5) for various total T_{prog}. For T_{prog}=50ns and 200ns, V_{switch} distributions were extrapolated from experiments at slower ramp speeds (higher total T_{prog}). (b) SET and RESET energy consumption distribution in RVS, comparing smart and non-smart procedure. (c) SET and RESET energy consumption distribution in CVS.

III. CONCLUSIONS

In this work we optimized RRAM programming consumption in kb array, including short latency and good endurance specifications. Targeting sufficient window margin, reducing programming time is more efficient than programming current reduction. Due to the large (>3 decades) intrinsic distribution of RRAM switching time, slow bits (@4σ) can exhibit programming energy 100 times higher than average behavior. Ramp voltage patterns allow to significantly reduce latency by compensating switching time variability with voltage increase, reducing consumption of several orders

of magnitude. Smart procedures with *on the fly* resistance measurements allow to reduce RESET latency time (1/2 decade) and consumption. Implementation of on-die smart circuit controls applied patterns. Mean consumption down to 1pJ in RESET and 500fJ in SET can be targeted.

ACKNOWLEDGMENT

This work has been partially supported by the European 621217 PANACHE project.

REFERENCES

[1] E. Shiu and S. Lim, "Driving Innovation in Memory Architecture of Consumer Hardware with Digital Photography and Machine Intelligence Use Cases *2017 IEEE International Memory Workshop (IMW)*, Monterey, CA, 2017, pp. 1-6.

[2] K. Prall, "Benchmarking and Metrics for Emerging Memory," *2017 IEEE International Memory Workshop (IMW)*, Monterey, CA, 2017, pp. 1-5.

[3] N. Raghavan *et al*., "Stochastic variability of vacancy filament configuration in ultra-thin dielectric RRAM and its impact on OFF-state reliability," *2013 IEEE International Electron Devices Meeting*, Washington, DC, 2013, pp. 21.1.1-21.1.4.

[4] C. Cagli *et al*., "Study of the Energy Consumption Optimization on RRAM Memory Array for SCM Applications," *2017 IEEE International Memory Workshop (IMW)*, Monterey, CA, 2017, pp. 1-4.

[5] A. Martina, P. O'Sullivan and A. Mathewsona, " Dielectric Reliability Measurements Methods: A Review, " *Microelectronics Reliability*, vol. 38, issue 1, pp. 37-72, Feb. 1998.

[6] W. C. Luo, K. L. Lin, J. J. Huang, C. L. Lee and T. H. Hou, "Rapid Prediction of RRAM RESET-State Disturb by Ramped Voltage Stress," *IEEE Electron Device Letters*, vol. 33, no. 4, pp. 597-599, April 2012.

[7] A. Kerber, L. Pantisano, A. Veloso, G. Groeseneken, M. Kerber, "Reliability screening of high-*k* dielectrics based on voltage ramp stress" *Microelectronics Reliability*, vol. 47, issue 4-5, pp. 513-517, April-May. 2007.

[8] A. Levisse, B. Giraud, J. P. Noël, M. Moreau and J. M. Portal, "Capacitor based SneakPath compensation circuit for transistor-less ReRAM architectures," *2016 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, Beijing, 2016, pp. 7-12.