A Novel Program-Verify Algorithm for Multi-bit Operation in HfO_2 RRAM

F. M. Puglisi, *Student Member, IEEE,* C. Wenger, and P. Pavan, *Senior Member, IEEE*

***Abstract*—In this letter we propose a dispersion-aware program-verify (DAPV) algorithm to enable reliable multi-bit operations in HfO_2-based RRAM. The significant intrinsic dispersion of the resistive states, typically hindering multi-bit operations, is exploited to devise a program-verify scheme which enables multi-bit operations with unique properties of failure resilience and adaptability to degradation. We show that an appropriate choice of the algorithm parameters can minimize the average number of cycles needed to program the cell, enabling fast and reliable multi-bit operation. This maximizes the bit/cell ratio and minimizes the dispersion of targeted resistive states.**

***Index Terms*—RRAM; Program-Verify; Variability; Resistive Switching, Multi-bit.**

I. Introduction

RESISTIVE Random Access Memory (RRAM) based on HfO_2 allows fast low-power operation, and high-density [1-2], becoming a promising alternative non-volatile memory. Still, inherent stochastic features, i.e. randomness in the switching mechanism [3-5], hamper multi-bit operations. Major concerns are cycling variability [3-4], i.e. considerable dispersion of High-Resistive (HRS) and Low-Resistive State (LRS) resistances, and resistance distributions degradation [6], i.e. changes in the stochastic response of the device over time. Controlling these phenomena, reducing RRAM reliability, requires employing program-verify (PV) schemes. In this letter, we propose a dispersion-aware PV (DAPV) scheme to minimize resistance dispersion and achieve reliable multi-bit operation. PV schemes for HfO_2 RRAMs reported in the literature, though efficient, are not tailored to prevent device failures [7-10]. Also, they are not designed to handle resistance distribution degradations. The proposed algorithm exploits the stochastic response of RRAM to the reset operation and the possibility of controlling the average HRS resistance via the reset pulse voltage to achieve unique properties of resilience to failures [11] and adaptability to degradation. The DAPV scheme enables 2 bit-cell operation, with better reliability compared to classical schemes [7-10].

II. Device and Experiments

TiN/10nm-Ti/10nm-HfO_2/TiN RRAMs are tested in 1T1R configuration – devices details are reported elsewhere [12]. The access nMOS is used to enforce a current compliance, I_C, to prevent current overshoot [13]. The preliminary forming operation is performed with I_C ranging from 50µA to 100µA by applying a DC voltage sweep up to 3.5V (not shown). Forming results in the creation of a Conductive Filament (CF) [3-5]. Then, we perform 50 complete pulsed switching cycles (set and reset), and we repeat the experiment for different reset conditions. The reset operation, driving the device in HRS, is performed by applying a negative voltage pulse ($-V_{RESET}$), resulting in a partial oxidation of the CF (creation of a dielectric barrier) [3-5]. The set operation, driving the device in LRS, is performed by applying a 2V pulse, and leads to the restoration of the CF [3-5]. The RRAM resistance, R, is measured after each operation by applying a V_{READ}=100mV pulse. Set, reset, and read pulses are all 10µs wide, including 2µs of rise/fall times. The total pulse period is 20µs.

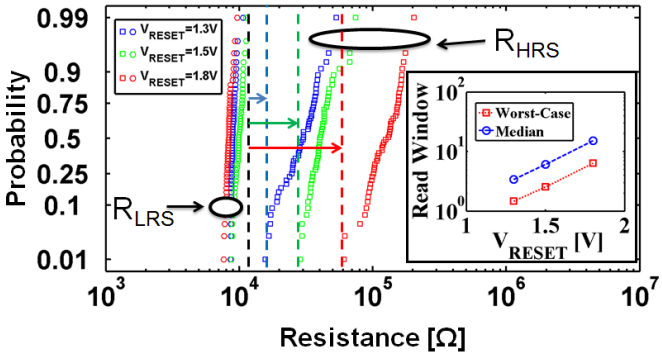

Figure 1. R_{LRS} and R_{HRS} distributions for three different V_{RESET}. I_C is 100µA. Only R_{HRS} distribution depends on V_{RESET}. The colored and black vertical dashed lines represent the minimum of each R_{HRS} distribution and the maximum R_{LRS}, respectively. The worst-case read window is indicated by an arrow for each V_{RESET}. In the inset, the worst-case (red squares) and the median (blue circles) read window vs. V_{RESET}.

III. The Program-Verify Algorithm

Figure 1 shows the probability distributions of both LRS and HRS resistances over 50 complete switching cycles when three different V_{RESET} are applied, namely 1.3V, 1.5V and 1.8V. The LRS resistance (R_{LRS}≈9kΩ) is normally distributed with no dependence on V_{RESET}, consistently with previous results and other studies [3-5, 11]. Conversely, the HRS resistance, R_{HRS}, is log-normally distributed [3-4] and strongly depends on V_{RESET}, resulting in a significant R_{HRS} dispersion. The unambiguous identification of the device state after each set/reset operation requires no overlap between the tails of the R_{LRS} and R_{HRS} distributions across the whole memory lifetime. Hence, a correct monitor of the device reliability is the worst-

Francesco Maria Puglisi, and Paolo Pavan are with the Dipartimento di Ingegneria "Enzo Ferrari", Università di Modena e Reggio Emilia, Via P. Vivarelli 10/1, 41125 Modena – Italy (phone: +39-059-2056324; fax: +39-059-2056329; e-mail: francescomaria.puglisi@unimore.it).

Christian Wenger is with IHP GmbH – Leibniz institute for innovative microelectronics, Im Technologiepark 25, 15236 Frankfurt (Oder), Germany.

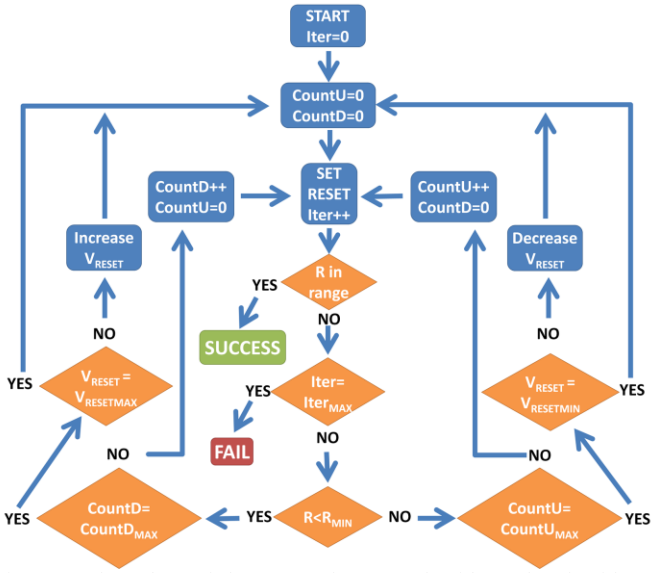

Figure 2. Flow chart of the proposed DAPV algorithm. The algorithm parameters are $Iter_{MAX}$, $CountU_{MAX}$, $CountD_{MAX}$ and the initial V_{RESET}. Orange boxes are comparison operation. Notably, only two comparisons per cycle are added, compared to the UPV.

case read window ($R_{HRS,MIN}/R_{LRS,MAX}$), much lower than the median read window ($R_{HRS,50\%}/R_{LRS,50\%}$) due to R_{HRS} dispersion, see the inset in Fig. 1. As such, the device shows reliable switching as, regardless of V_{RESET}, the R_{LRS} and R_{HRS} distributions are never overlapped, Fig. 1. Unfortunately, the R_{HRS} distributions related to the three different V_{RESET} are mostly overlapped. This hinders multi-bit operation, which can be achieved only by applying PV schemes [7-10]. Typically, it consists in delivering to the device a train of pulses with incrementally higher voltage or pulse width during reset (program), until the desired R level (verify) is reached [7-10]. In general, if the desired condition is not met within 20 pulses the algorithm fails [7-10]. However, this procedure cannot be reliably applied to TiN/Ti/HfO_2/TiN RRAM, since it may induce failures. Indeed, the first reset pulse creates a dielectric barrier within the CF [3-5]. The high electric field given by the following pulse may lead to the barrier disruption ("negative set failure") [11], rather than to an R_{HRS} increase. This failure mode occurs when the reset operation is performed on a device which is already in HRS. This scenario is associated with two competing physical mechanisms: 1) the further growth of the dielectric barrier within the CF and 2) the breakdown of the barrier due to the applied electric field. So, the application of consecutive reset pulses may cause the device to fail. Hence, different strategies have to be devised to achieve "negative set failure" resilience.

A. *Unconstrained Program-Verify (UPV)*

A straightforward method to target a tight R_{HRS} distribution (from R_{MIN} to R_{MAX}) while avoiding the "negative set failure" would be the following:

1. Choose V_{RESET} and a target R span (R_{MIN} - R_{MAX})
2. Reset the device
3. Check if the resistance, R, is R_{MIN}<R<R_{MAX}
4. If not, apply a set pulse and go back to 2

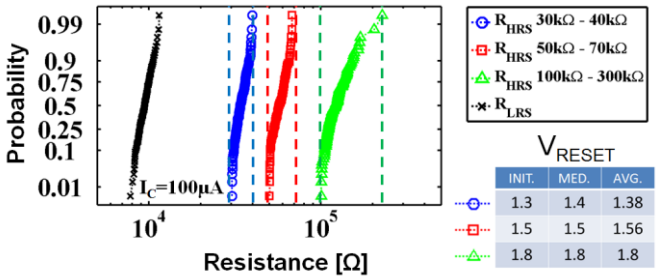

Figure 3. R_{LRS} and three R_{HRS} distributions for 100 cycles as obtained with DAPV. For each target R span, the initial V_{RESET} is varied by the algorithm to ensure successful writing and minimal programming time. The table reports the initial V_{RESET} for the three target R_{HRS} span, along with the median and average values over 100 algorithm runs of the effective V_{RESET} which is used by the algorithm when succeeding in programming. The other parameters are set to $Iter_{MAX}$=20, $CountU_{MAX}$=$CountD_{MAX}$=3.

Notably, in classical schemes the set pulse is applied only when R>R_{MAX} [7-10], whereas another reset pulse is applied if R<R_{MIN}. Here, instead, a set pulse is applied every time the programmed R value is out of the target R span. This approach is more time-consuming due to the additional set pulse but circumvents the "negative set failure". However it requires an accurate matching between the applied V_{RESET} and the selected R span. For instance, Fig. 1 suggests it is unlikely to achieve $10^5\Omega$<R<$2\cdot10^5\Omega$ when V_{RESET}=1.3V. This choice will imply the algorithm failure (missing the target R span within 20 cycles). So, the algorithm parameters (V_{RESET}, R_{MIN}, and R_{MAX}) must be wisely chosen to guarantee success and to minimize total programming time:

$$T_{PROGARM} = n(T_{RESET} + T_{SET} + T_{CHECK}) \qquad (1)$$

with n being the number of cycles, T_{RESET} (T_{SET}) the reset (set) pulse period and T_{CHECK} the time to verify if R is within the targeted R span. However, if the R_{HRS} distribution changes during device operation [6], even a good initial choice of the parameters may cause the algorithm to fail. As such, this scheme is resilient to the "negative set failure" but shows no adaptability to degradation.

B. *Dispersion-Aware Program-Verify (DAPV)*

A more robust approach is based on the real-time estimation of the most appropriate V_{RESET} for the targeted R span. Indeed the R span which can in fact be targeted in the reset operation depends on V_{RESET}, Fig. 1. If the algorithm misses the target R span by *always* exceeding (falling behind) it, then reducing (increasing) V_{RESET} is required to achieve the target R span. The proposed scheme is summarized in the flow chart in Fig. 2. The counter "Iter" tracks the number of cycles, while "CountU" and "CountD" track how many *consecutive* times the programmed R exceeds (falls behind) the target R span. After each program operation (set and reset) and R evaluation, this scheme checks if V_{RESET} must be changed for the next iteration. To this point, either CountU or CountD is compared to its maximum value, which is a parameter of the algorithm as well as the initial V_{RESET}, R_{MIN}, R_{MAX}, and the maximum number of cycles $Iter_{MAX}$ (typically 20). Moreover, we also set

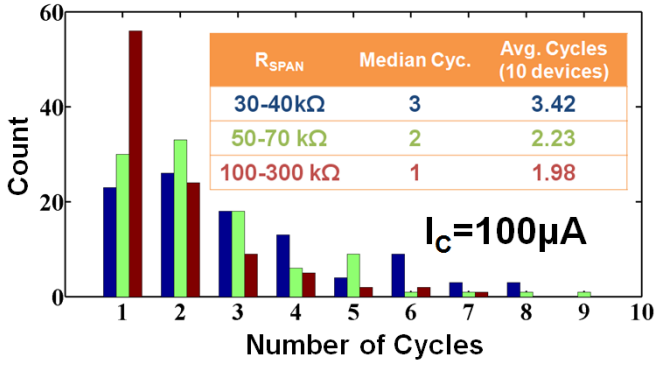

Figure 4. DAPV performance evaluated on a device formed with I_C=100µA (R distributions in Fig. 3). Regardless the target R span, DAPV succeeds in < 10 cycles and shows no failures achieving two bits/cell operation. The table (inset) reports the median number of iterations needed for DAPV to succeed for the device under test and the same value averaged over 10 devices tested in the same conditions.

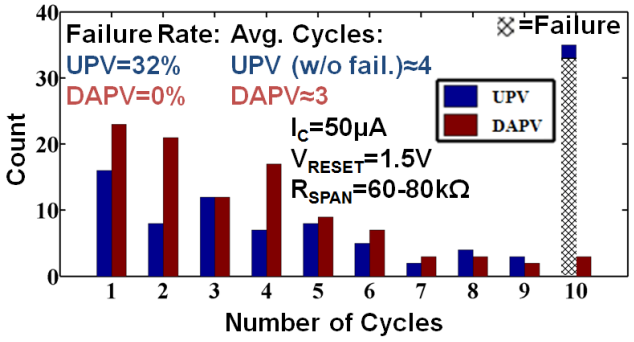

Figure 5. Comparison between UPV and DAPV on a device formed with I_C=50µA. UPV is run 100 times with V_{RESET}=1.5V. DAPV is run 100 times with $CountU_{MAX}$=$CountD_{MAX}$=3 and initial V_{RESET}=1.5V. $Iter_{MAX}$=10 for both schemes. UPV failure rate is 32%, while DAPV never fails. The average number of cycles to succeed ≈ 3 for DAPV, and ≈4 for UPV (without considering failures).

boundaries on V_{RESET} to prevent over-reset and under-reset failures ($V_{RESET,MIN}$=1.3V, $V_{RESET,MAX}$=1.8V) [11]. The DAPV scheme allows optimizing the reset conditions according to the actual device statistical response [3-5], showing adaptability to R_{HRS} distributions degradation [6].

IV. Discussion

The proposed algorithm is tested 100 times with different input parameters combinations, targeting three different R spans. R_{MIN}, R_{MAX} and initial V_{RESET} values for each target R span are reported in Fig. 3. The remaining parameters are fixed at $CountU_{MAX}$=$CountD_{MAX}$=3 (chosen to minimize the average number of cycles required to program the device - not shown) and $Iter_{MAX}$=20. Resistance distributions are reported in Fig. 3 showing the superior control of the final resistance dispersion. Fig. 4 shows the performance metric of the DAPV. The algorithm always succeeds in < 10 cycles, and performs better for larger R_{HRS} target spans, as expected. This confirms the possibility of reliable two-bits/cell operation with no failures. Similar performances are obtained on 10 different devices, as reported in the inset of Fig. 4. Notably, the user can freely define the number of target R spans and their boundaries (distributions margins), trading-off between the number of bits/cell and algorithm performance. Compared to UPV, DAPV offers much better performance. Fig. 5 indicates that DAPV can program a cell in < 10 cycles with no failures, while UPV failure rate in the same conditions is 32%. Also, DAPV has a better success rate at low cycles than UPV, which reduces the possible endurance degradation resulting from the application of program-verify schemes. So, though being computationally heavier than UPV due to higher complexity (compared to UPV, two comparisons per cycle are added, Fig. 2), the DAPV shows lower average programming time.

V. Conclusions

We proposed a new dispersion-aware program-verify algorithm for HfO_2 RRAM. It exploits the intrinsically stochastic response of RRAMs to achieve an optimal control of resistance values. It allows reliable multi-bit operation with no failures and optimized programming time, while displaying the unique properties of failure resilience and adaptability to degradation.

References

[1] Y.-S. Chen et al., "Highly scalable hafnium oxide memory with improvements of resistive distribution and read disturb immunity", Proc. of IEEE Electron Devices Meeting (IEDM) 2009, pp.1,4, 7-9 Dec. 2009.

[2] H.-Y. Lee et al., "Low-power switching of nonvolatile resistive memory using hafnium oxide", Jpn. J. Appl. Phys., vol. 46, no. 4B, pp. 2175–2179, 2007.

[3] F. M. Puglisi, P. Pavan, A. Padovani, and L. Larcher, "A Compact Model for Hafnium-Oxide-Based Resistive Random Access Memory", Proceedings of the International Conference on IC Design & Technology (ICICDT), pp.85-88, 29-31 May 2013.

[4] L. Larcher, F. M. Puglisi, P. Pavan, A. Padovani, L. Vandelli, and G. Bersuker, "A Compact Model of Program Window in HfO_x RRAM Devices for Conductive Filament Characteristics Analysis", IEEE Transactions on Electron Devices, vol.61, no.8, pp.2668-2673, Aug. 2014.

[5] F. M. Puglisi, L. Larcher, G. Bersuker, A. Padovani, and P. Pavan, "An Empirical Model for RRAM Resistance in Low- and High-Resistance States", IEEE Electron Device Letters, vol.34, no.3, pp.387-389, March 2013.

[6] S. Deora et al., "Statistical assessment of endurance degradation in high and low resistive states of the HfO_2-based RRAM", IEEE International Reliability Physics Symposium (IRPS) 2013, pp.MY.2.1-MY.2.5, 14-18 April 2013.

[7] G. Wang et al., "Impact of program/erase operation on the performances of oxide-based resistive switching memory", Nanoscale Research Letters (2015) 10:39.

[8] Y.S. Chen et al., "Highly scalable hafnium oxide memory with improvements of resistive distribution and read disturb immunity", IEEE International Electron Devices Meeting (IEDM), pp.1,4, 7-9 Dec. 2009

[9] K. Higuchi, T.O. Iwasaki, and K. Takeuchi, "Investigation of Verify-Programming Methods to Achieve 10 Million Cycles for 50nm HfO_2 ReRAM", 2012 4th IEEE International Memory Workshop (IMW), pp.1-4, 20-23 May 2012.

[10] S. Ning, T. O. Iwasaki, and K. Takeuchi, "50 nm Al_xO_y resistive random access memory array program bit error reduction and high temperature operation", 2014 Jpn. J. Appl. Phys., vol. 53, 04ED09.

[11] S. Balatti et al., "Pulsed cycling operation and endurance failure of metal-oxide resistive (RRAM)," IEEE International Electron Devices Meeting (IEDM) 2014, pp.14.3.1-14.3.4, 15-17 Dec. 2014.

[12] D. Walczyk et al., "Resistive switching behavior in TiN/HfO_2/Ti/TiN devices", International Semiconductor Conference Dresden-Grenoble (ISCDG), 2012, pp.143-146, 24-26 Sept. 2012.

[13] A. Chen, "Current overshoot during set and reset operation of resistive switching memories", IEEE International Reliability Physics Symposium (IRPS) 2012, pp.MY.2.1-MY.2.4, 15-19 April 2012.