Experimental Fault Rate Characterization and Protection in Embedded RRAM

Connor Talley[1], Brian Crafton[1], Samuel D. Spetalnick[1], Muya Chang[1], and Arijit Raychowdhury[1]
[1]Georgia Institute of Technology
Email: arijit.raychowdhury@ece.gatech.edu

2022 Opportunity Research Scholars Symposium (ORSS) | 978-1-6654-6898-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ORSS55359.2022.9806038

***Abstract*—Resistive Random Access Memory (RRAM) is a process and voltage compatible embedded nonvolatile memory with high density. Embedded non-volatile memory has gained interest as a dense embedded memory consuming near-zero leakage power and enabling higher on-chip capacity. These properties are particularly interesting for overcoming the main-memory bottleneck in resource-constrained systems. Unfortunately, RRAM faces new challenges traditional charge-based memories have avoided. In this work, we characterize the impact of faults in RRAM and explore various forming procedures to mitigate high fault rates. We implement two error correction techniques to eliminate fault-induced errors and evaluate area overhead. We demonstrate how the combination of these methods can achieve a 1516.4 times reduction in faults over prior work.**

I. Introduction

With each year, the dependence of cutting edge computing systems on speed, capacity, and bandwidth of memory systems increases [1]. Additionally, processes heavily dependent on bandwidth and on-chip capacity such as machine learning (ML) are seeing a steady climb in use [2]. To address these demands, large investments into researching both hardware accelerators [3] and software frameworks have been made. Due to these efforts, we have seen significant improvement in both the performance and energy efficiency in memory systems comprised of DRAM and/or SRAM. However, a significant drawback of SRAM is its area cost, with it often consuming half of chip area [4]. DRAM can be used to compensate for this lack of density, but has high energy demand [5]. DRAM is also implemented as an off-chip module, but this leads to a greater amount of energy being dedicated to data movement [6]. The performance limitations and energy demands of DRAM and SRAM, combined with a decline of Moore's law, have inspired demand for new memory technology and techniques to address the future demands of computing systems.

Fortunately, research promising to push the bounds of what is possible with CMOS technology is already underway. Compute in-memory (CIM) is one such research thread that reads and accumulates multiple memory cells onto the same bitline (BL), allowing (binary) multiplication and addition without the use of CMOS logic. At the same time, embedded non-volatile memory (eNVM) such as resistive RAM (RRAM) and phase change RAM (PCRAM) are making strides towards commercial viability [1]. Memories such as RRAM and PCRAM offer high density non-volatile storage, having similar construction and density to DRAM while consuming only a fraction of the power. RRAM and PCRAM are also process and voltage compatible with current technology, aiding in ease of integration. Furthermore, these technologies store information through change of resistance which can enable multi-level storage and a more natural primitive for compute in-memory. However, as RRAM and PCRAM are still in development, there are many features about these memory technologies that still need to be characterized, such as fault rate.

In this work, we evaluate various approaches to minimizing hard failures occurring during the initial formation of RRAM cells. To experimentally quantify fault rate, we use a 40nm foundry RRAM test chip to characterize RRAM cells [7], [8]. The chip contains 288 256×256 RRAM arrays (18.9 Mb) and an on-chip CPU for fast data collection. The details of the read and write circuit of the chip are beyond the scope of this paper and interested readers are pointed to [7], [8] for further discussions.

Next, we performed an evaluation to determine the cell formation parameters that lead to the lowest overall failure rate. We demonstrate a 9.9 times reduction in cells experiencing hard failures from a baseline set of parameters, achieving the lowest fault rate in literature to the best of our knowledge. We then combine this with the ECP protocol defined in [9]. Compared to previous works, an overall fault reduction of 1516.4 times is achieved [10].

II. Background and Motivation

A. Structure and Operation of Resistive RAM (RRAM)

RRAM (often called ReRAM) is a filamentary device that switches between a high resistance state (HRS) and low resistance state (LRS) based on the direction of current applied across the two terminals. The HRS and LRS in RRAM are achieved by forming and destroying a filament inside the insulator material of the device. By creating and destroying this filament, we can lower and raise the resistance of the device by orders of magnitude. The transition from HRS to LRS is called the *set* process where the device allows more current to flow, emulating a digital '1'. The transition from LRS to HRS is called the *reset* process where the device is

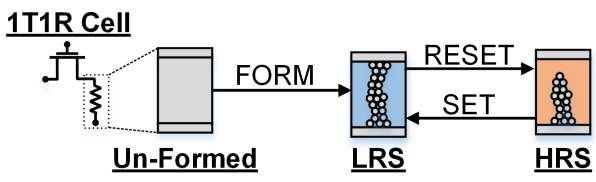

Fig. 1. Architecture of 1T1R RRAM cell. RRAM operates through operations creating and destroying a filament.

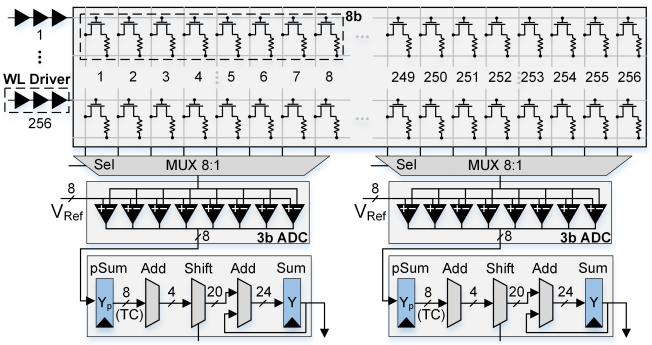

Fig. 2. 256×256 RRAM array architecture. 8 adjacent cells share a 3-bit ADC through a 8-to-1 multiplexer. Local shift-and-add logic enables near memory compute.

less conductive and results in less current across the terminals, emulating a digital '0'. Since a read and write operation both apply voltage on the two terminals, the read voltage must be much lower to not alter the state of the device and perform a destructive read. In the 1T1R (1 transistor, 1 resistor) structure, the read voltage is controlled by using a small voltage on the gate of the transistor.

Although there are different types of RRAM, the most successful is metal-oxide RRAM [11]. The device structure of metal-oxide RRAM simple, comprised of a top metal electrode, a bottom metal electrode, and a transition metal oxide layer (TMO) in-between, as shown in Figure 1. RRAM initially starts in a pristine state, and most devices must undergo forming prior to being used as intended. During formation, an initial large voltage is required to create an electric field capable of knocking oxygen atoms out of the insulator's lattice and creating vacancies that make up the conductive filament leading to the LRS. The forming process only needs to be done once, taking the device from its initial pristine state, which has a resistance orders of magnitude larger than the HRS of the device post formation.

When considering the formation process in RRAM, the main parameters to consider are the write voltage (forming voltage) and the length of its application (pulse width). The formation of an RRAM cell can be thought of as power-dependent. Using a slightly lower voltage during the forming process of RRAM cells can sometimes be achieved through a longer pulse width, demonstrating an interdependence between forming voltage and pulse width.

B. Formation Faults In RRAM

AS detailed previously, the primary purpose of the formation of RRAM cells is to create the filament of the cell and as a result, lower the resistance to LRS. However, the target resistance of LRS is not always achieved at the end of formation. In most cases, these cells that fail to achieve LRS become unresponsive to future write operations. RRAM cells that become unresponsive following formation can be placed in one of two categories of formation faults, underformed and overformed. Underformation occurs when an RRAM cell's resistance is insufficiently lowered, leading to subsequent set operations failing to achieve LRS. In the case of an RRAM cell's resistance being lowered too much (below LRS), over-formation occurs, and standard reset operations are incapable of raising the resistance of the cell to HRS. The possibility to overform and underform RRAM cells adds a degree of complexity to achieving lower fault rates post-formation, as it places both lower and upper limits on what the forming voltage and pulse width can be.

C. Fault Detection and Correction

All modern memory systems suffer from hard faults as well as transient errors. Depending on the type of memory and its application, these memories utilize various forms of error correction codes (ECC). The various forms of ECC utilize "check" bits based on information theory to detect, localize, and correct errors when they occur. These check bits used require significant overhead that can take away a significant amount of usable memory. This reveals an important trade-off between reliability and area overhead that must be considered when choosing an ECC. In Section IV, we explore this trade-off by analyzing the area overhead and fault rates when two different ECCs are applied to RRAM.

III. Experimental Characterization

In the previous section, the need to carefully consider both parameters (form voltage, pulse width) used in the formation of RRAM cells was identified. It is impractical to examine every possible combination of the two parameters to establish

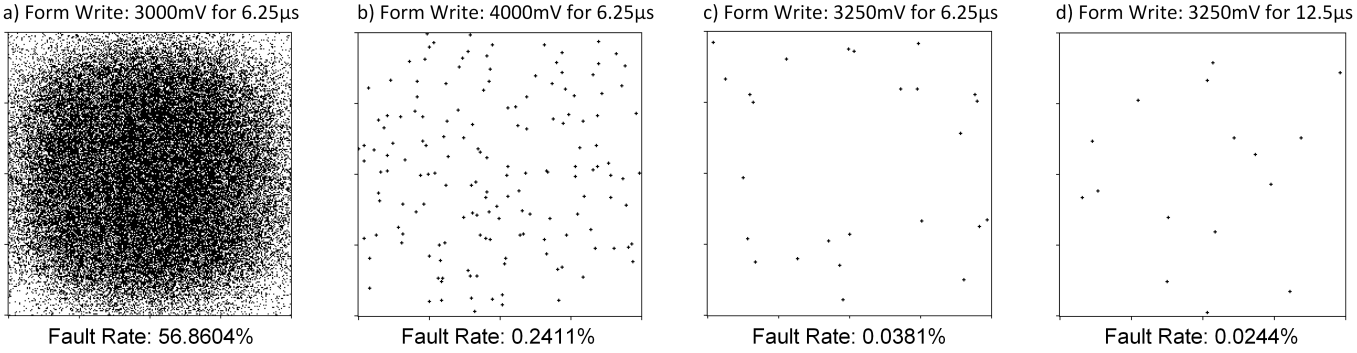

Fig. 3. Fault maps of 256x256 arrays of RRAM cells. a) demonstrates insufficient formation power. b) represents sufficient formation power before the application of form optimization. c) results from use of voltage optimization only, while d) is achieved through combined use of voltage and pulse width optimization.

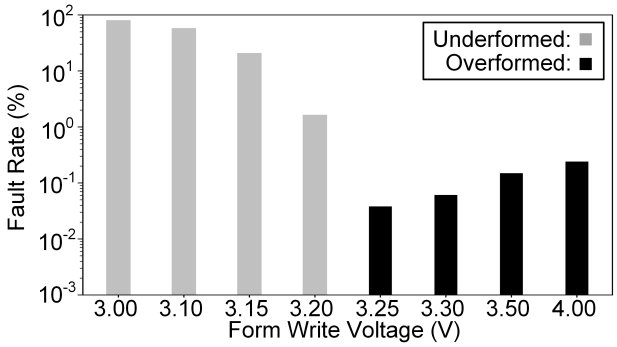

Fig. 4. Characterization of Forming Voltage effect on Stuck-At Fault rate. Pulse width is held constant at 6.25µs for every voltage.

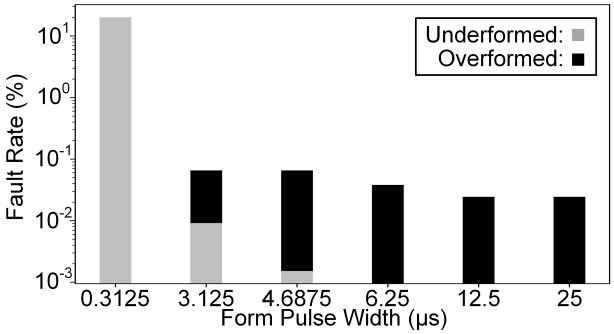

Fig. 5. Characterization of Pulse Width effect on Stuck-At Fault rate. Forming voltage is held constant at 3.25V for every pulse width.

the optimal formation configuration. As such, first, the effects of the forming voltage on the fault rate are evaluated while holding the pulse width constant. Then, the effects of varying the pulse width are observed with the forming voltage set to the optimal (lowest fault rate) value found in the voltage testing. This method provides an effective way to quickly approximate the optimal formation configuration.

The formation configurations used in the collection of the data presented over the next few sections are as follows:

1) *Form Voltage*:
   3.0V, 3.1V, 3.15V, 3.20V, 3.25V, 3.3V, 3.5V, 4.0V
2) *Form Pulse*:
   0.3125µs, 3.125µs, 4.6875µs, 6.25µs, 12.5µs, 25µs

A. Forming Voltage Optimization

To determine the optimal forming voltage, we first aim to define the minimum voltage required to eliminate all underforming faults. This fits in with the earlier assumption that above the optimal voltage range, we will see an increase in the rate of overformed/stuck cells, while underformed cells appear below the optimal range. The relationship between voltage and stuck cells of any type does not follow a simple parabolic curve. This is seen in the 3.00V-3.20V range in Figure 4, where the percentage of underformed cells increases significantly with a slight decrease in voltage. Alternatively, in the 3.25V-4.00V range in Figure 4, we observe that while increasing forming voltage does lead to an increase in the number of overformed cells, this increase is less than was found for underforming. As such, we propose that when approximating the optimal forming voltage, minimizing the number of underformed cells first will lead to consistently lower fault rates.

In our experiment, we formed 256×256 RRAM arrays (65,536 cells) at increments of 100mV. We also decreased this step size to 50mV around the region in which the primary fault switches from underformed to overformed, as this feature contains the minimum fault rate. We determined through this process that the optimal forming voltage out of the ones tested for our RRAM test chip is 3.25V, as can be seen in the results of Figure 4.

B. Pulse Width Optimization

The process for determining the optimal pulse width at the optimal voltage is similar to how the optimal voltage is found, with the minimization of underformation also taking priority. However, as can be seen in Figure 5, greatly increasing the pulse width past what is required to minimize underforming does not appear to have a significant effect on the overforming of cells. As can be seen in Figure 5, the rate of unresponsive cells appears to decrease exponentially as the pulse width increases, meaning that instead of there being an apparent optimal pulse width, there is a region of diminishing returns starting at 6.25µs. This explains why there is no change in the rate of unresponsive cells when the pulse width is doubled from 12.5µs to 25µs. As such, we propose that a pulse width of 12.5µs is sufficient in obtaining a minimum fault rate of 0.0244% on our test chip.

IV. Fault Detection and Correction

A. SECDED and ECP for Fault Protection

To overcome both transient bit errors (soft errors) and faults (hard errors) commercial memory systems use various forms of ECC. ECC has found widespread use in all levels of the memory hierarchy ranging from SRAM to DRAM to Flash. ECC typically comes with significant area overhead to store the check bits used for error correction. Because these check-bits reduce effective memory capacity, the choice of ECC scheme must be made to ensure target bit error rate while minimizing area overhead.

To evaluate the necessary ECC overhead for RRAM, we consider two common ECC schemes: *Single Error Correction*

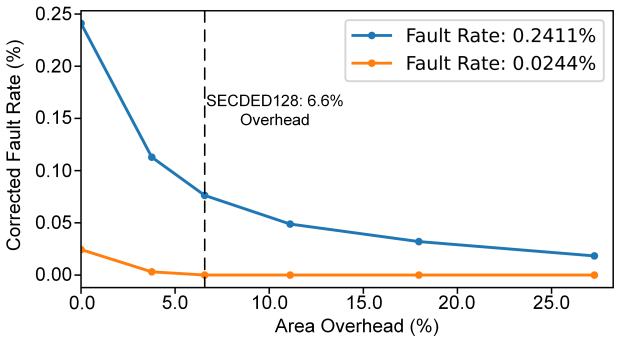

Fig. 6. Corrected fault rate achieved by SECDED with a given area constraint. Here the results of Figure 3 b) are compared against Figure 3 d).

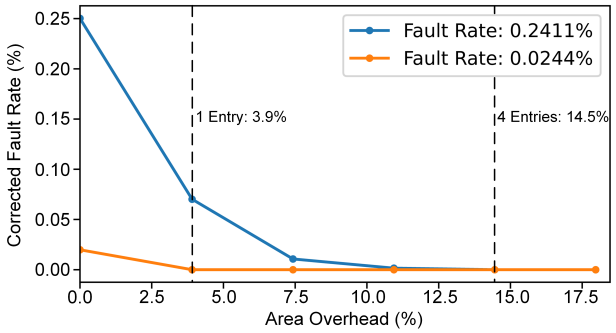

Fig. 7. Corrected fault rate achieved by ECP with a given area constraint. Here the results of Figure 3 b) are compared against Figure 3 d).

Double Error Detection (SECDED) and *Error Correction Pointer* (ECP). SECDED is a classic ECC scheme used in memory systems today. It is based on hamming or BCH codes and pads check bits to arbitrary data blocks that can be used to detect two errors and correct one. ECP [9] is a more recent scheme that appends pointers in rows of the memory array and is intended for use in PCRAM. These pointers store addresses to faulty cells, so that upon reading the row of the array, we can correct known fault. Due to the similarities of PCRAM and RRAM, implementing ECP in RRAM should also be effective. One major difference between SECDED and ECP is that SECDED can correct both transient errors and hard faults while ECP can only correct hard faults. However, ECP yields lower error rates for similar area overhead when only hard faults are considered.

B. Simulation Results

To evaluate the two ECC schemes, we simulate their performance and overhead on our fault measurements. Performance is measured as fault rate after the ECC is applied (i.e. corrected fault rate), and overhead is the area cost of the scheme. In Figure 6, we show the results for our simulations using SECDED schemes with different area overheads. We apply SECDED to arbitrary sized data blocks to explore the fault rate to overhead tradeoff. A higher area overhead is required when SECDED is applied to fewer bits. For instance, if SECDED is applied to every 16 bits, each data block requires 6 check bits, and thus requires 37.5% overhead. If instead, SECDED is applied to every 256 bits, each data block requires 10 check bits, and thus requires 3.9% overhead. As can be seen in Figure 6, SECDED128, with an overhead of 6.6%, is necessary to have a corrected fault rate of 0% at a base (without ECC) fault rate of 0.0244% (the lowest rate found in testing), while an overhead of over 37.5% is needed at the baseline of 0.2411%.

In Figure 7, we show the results for our simulations using ECP. To explore the trade-off between area and fault rate, we try different numbers of pointers per RRAM row (256 bits), with a maximum of 5 pointers (ECP_5) Naturally, more pointers achieves less faults but comes at the cost of higher area. As can be seen in Figure 7, ECP_1 at 3.9% overhead is necessary to have a corrected fault rate of 0% at a base (without ECP) fault rate of 0.0244%.At the baseline of 0.2411% ECP_4 at 14.5% overhead, is needed. Comparing ECC to ECP in the case of a base fault rate of 0.0244%, we see a 1.7 times reduction in necessary area overhead. From these experiments, we conclude that ECP is better suited to RRAM when only faults are considered. However, as the RRAM technology matures and fault rates lower, SECDED will become more advantageous because of its ability to detect and correct transient errors.

V. Conclusion

By optimizing the voltage and pulse width used to form RRAM cells, up to a 10 times reduction in fault rate is obtainable. This allows RRAM to be used and tested more efficiently, especially in CIM ML applications that require the use of multiple adjacent RRAM cells. We determined that underformation faults are much more sensitive to changes in forming voltage and pulse width than overformation faults, and that the forming parameters chosen should prioritize eliminating underformation faults first. We achieved an overall lowest fault rate of 0.0244%, the lowest rate found in literature, to the best of our knowledge, by 1516.4 times. When combined with ECC to achieve a 0% corrected fault rate, required area overhead dedicated to ECC is reduced by as much as 4-6 times.

References

[1] B. Crafton *et al.*, "Merged logic and memory fabrics for accelerating machine learning workloads," *IEEE Design & Test*, 2020.

[2] V. Sze *et al.*, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[3] N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, pp. 1–12, IEEE, 2017.

[4] T. Singh, S. Rangarajan, D. John, R. Schreiber, S. Oliver, R. Seahra, and A. Schaefer, "2.1 zen 2: The amd 7nm energy-efficient high-performance x86-64 microprocessor core," in *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, pp. 42–44, 2020.

[5] S. Ghose, A. G. Yaglikçi, R. Gupta, D. Lee, K. Kudrolli, W. X. Liu, H. Hassan, K. K. Chang, N. Chatterjee, A. Agrawal, M. O'Connor, and O. Mutlu, "What your dram power models are not telling you: Lessons from a detailed experimental study," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 2, dec 2018.

[6] A. Boroumand, S. Ghose, Y. Kim, R. Ausavarungnirun, E. Shiu, R. Thakur, D. Kim, A. Kuusela, A. Knies, P. Ranganathan, and O. Mutlu, "Google workloads for consumer devices: Mitigating data movement bottlenecks," *SIGPLAN Not.*, vol. 53, p. 316–331, mar 2018.

[7] S. D. Spetalnick, M. Chang, B. Crafton, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, and A. Raychowdhury, "A 40nm 64kb 26.56 tops/w 2.37 mb/mm 2 rram binary/compute-in-memory macro with 4.23 x improvement in density and¿ 75% use of sensing dynamic range," in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 65, pp. 1–3, IEEE, 2022.

[8] M. Chang, S. D. Spetalnick, B. Crafton, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, and A. Raychowdhury, "A 40nm 60.64 tops/w ecc-capable compute-in-memory/digital 2.25 mb/768kb rram/sram system with embedded cortex m3 microprocessor for edge recommendation systems," in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 65, pp. 1–3, IEEE, 2022.

[9] S. Schechter, G. H. Loh, K. Strauss, and D. Burger, "Use ecp, not ecc, for hard failures in resistive memories," *SIGARCH Comput. Archit. News*, vol. 38, p. 141–152, jun 2010.

[10] C.-Y. Chen, H.-C. Shih, C.-W. Wu, C.-H. Lin, P.-F. Chiu, S.-S. Sheu, and F. T. Chen, "Rram defect modeling and failure analysis based on march test and a novel squeeze-search scheme," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 180–190, 2015.

[11] H.-S. P. Wong, H.-Y. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, B. Lee, F. T. Chen, and M.-J. Tsai, "Metal–oxide rram," *Proceedings of the IEEE*, vol. 100, no. 6, pp. 1951–1970, 2012.