Testing Resistive Memory based Neuromorphic Architectures using Reference Trimming

Christopher Münch
Department of Computer Science
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
christopher.muench@kit.edu

Mehdi B. Tahoori
Department of Computer Science
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
mehdi.tahoori@kit.edu

***Abstract*—Neuromorphic architectures based on emerging resistive memories are in the spotlight of today's research as they are able to solve complex problems with an unmatched efficiency. In particular, resistive approaches offer multiple advantages over CMOS-based designs. Most prominently they are non-volatile and offer small device footprints in addition to very low power operation. However, regular memory testing used for conventional resistive Random Access Memory (RAM) architectures cannot detect all possible faults in the synaptic operations done in a resistive neuromorphic architecture. At the same time, testing all neuromorphic operations from the logic testing perspective is infeasible. In this paper we propose to use reference resistance trimming for the test phase and derive a generic test sequence to detect all the faults impacting the neuromorphic operations based on an extensive defect injection analysis. By exploiting the resistive nature of the underlying architecture, we are able to reduce the testing time from an exponential complexity necessary for a conventional logic testing approach to a linear complexity and reduce this by another 50% with the help of resistance trimming.**

***Index Terms*—In-memory computing, Memory Testing, resistive memories, resistance trimming**

I. Introduction

Problem solving has always been the main driver for innovation and progress. With an increase in problem complexity, new ways are explored to efficiently solve them. In recent years, machine learning in general and brain inspired neuromorphic computing specifically have been gaining ever increasing attention from the scientific community. Using the neuromorphic computing paradigm, it is possible to directly implement efficient Artificial Neural Networks (ANNs) to perform even the most complex tasks, which were previously not efficiently solvable by conventional computing architectures. These ANNs have a high memory and computational demand and hence profit from dedicated hardware specifically tailored to their needs.

Emerging resistive memories, like Resistive Random Access Memory (ReRAM) [1], Phase Change RAM (PCRAM) [2] or Spin Transfer Torque Magnetic RAM (STT-MRAM) [3] offer new and innovative ways to combine this demand of computation and memory in form of Compute in-Memory (CiM) architectures [4]. The use of resistive CiM architectures allows the implementation of efficient ANNs. The synaptic weights of an ANN can be stored directly in a non-volatile and highly dense way, which is of particular interest for low power portable devices. Also, the CiM-capabilities of resistive cells offer a convenient way to perform the inference of the ANN directly in-memory by summing up and evaluating the current of multiple activated cells.

Whilst all of the emerging resistive memories offer a similar set of features, like non-volatility and varying resistive states, their specific characteristics differ a lot. As they use different physical concepts to realize their resistive behavior, characteristic features like the differences of their low resistive states (LRS) to their high resistive states (HRS), their endurance and their write operation delay and energy demand change from one technology to another. Their manufacturing processes are subject to new and different failure modes and mechanisms compared to the conventional CMOS process. A smaller LRS/HRS ratio of a device has a direct impact on the sensibility of the implemented functionality and thus the manifestation of defects into ANN faults.

In this paper, we elaborate on the possible defects and faults of CiM-based ANN hardware, specifically their influence on threshold operations needed for neuromorphic operations. We show the exploitation of the resistive cell behavior to reduce the testing time significantly compared to pure logic testing. Additionally, we take advantage of resistance trimming, to reduce the write operations during the test sequence even further. We chose STT-MRAM to evaluate our resistive testing approach, as it is one of the more mature technologies with a comparable low LRS/HRS ratio and thus the most vulnerable technology to defects which can cause failures in neuromorphic hardware. By selecting the most demanding technology, we are able to generalize our extensive defect injection simulation based analysis to the other more relaxed memory types. These resistive defects typically result in faulty behavior of the cell during a neuromorphic operation. Conventional memory tests are not sufficient to spot the CiM-specific faults [5], the same is therefore true for CiM-based neuromorphic memories, used to implement efficient ANN hardware.

The rest of the paper is organized as follows. In Section II we discuss the relevant related work. Section III gives our general simulation approach and results followed by Section IV with the discussion of our baseline and the derivation of our proposed test generations. Section V concludes our work.

978-3-9819263-5-4/DATE21/©2021 EDAA

II. Related Work

Testing of emerging resistive memory technologies are subject to many studies. Specifically testing of conventional memory operations is discussed in [6]–[9]. Some of them focus on a single technology, e.g. STT-MRAM [6] or RRAM [7], [8], whereas others have a broader view on multiple resistive memory technologies [9]. Testing of CiM-capable architectures is discussed in [10] with extensive fault modeling, an RRAM core array and a focus on the memory periphery. STT-MRAM specific defect modeling and test generation in presented in [5]. However, there are also pure CMOS based CiM design: In [9] the testing of an 8T SRAM cell for in-memory computing is presented. Also there are test concepts to realize CiM operations with resistive memories but without the conventional RAM architecture, e.g. Memristor Rationed Logic [11]. In [12] the general approach of testing the functionality of a neuromorphic circuit is compared to the sole structural testing of the circuitry.

As neuromorphic architectures are very different to regular memories and can heavily utilize CiM-operations, it is necessary to specifically test them and potentially optimize these tests for this neuromorphic scenario. None of the previous work have addressed the specific need of test generation for CiM-based neuromorphic architectures. We therefore show the derivation of a test generation to detect all (single) faults in a CiM-based neuromorphic memory architecture. This is done by extensive fault injection evaluation and re-purposing the resistive trimming circuit.

III. Fault Analysis and Test Requirements for Resistive Neuromorphic Hardware

In this section we have a closer look at the test requirements for neuromorphic hardware, specifically the implications of using non-volatile resistive memory technologies for their implementation. We also use STT-MRAM as a proof of concept because it is the most sensible technology in terms of the sense margin due to a very low LRS/HRS ratio. To derive our proposed test sequences we use extensive technology-aware defect injection simulations. For illustration purposes, we use the threshold operations with four inputs and generalize to the full array afterwards to derive the test sequence. As write faults can easily be handled by conventional memory test pattern, we focus on the testing of synaptic operation faults (during inference) which are different from conventional memory read faults [13], [14].

We performed all of our simulations with *Cadence Virtuoso* and *Cadence Spectre*. Our fault injection framework is described in detail in [13]. The TSMC 40nm Low Power SPICE models were used to model the CMOS devices of our evaluated architecture. Furthermore, an MTJ compact model as described in [15] was used as the resistive memory cell with a free layer/oxide thickness of 1.3/1.48 nm, a RA-product of 7.5 $\Omega\mu m^2$ and a TMR of 150% resulting in a 'AP'/'P' resistance of 15 kΩ/6 kΩ. We performed our simulation with a circuit supply voltage of 1.2V and a nominal temperature of 27°C.

A. Neuromorphic Test Requirements

Depending on the operand configuration, the sense margin to differentiate between the states of the activated memory cells changes. As shown in Figure 1 for four cells, the difference between reference resistance R_{MIN1} and the "0000"/"1000" states (SM1) is significantly smaller than the difference between R_{MIN4} and the "1110"/"1111" states (SM4). To generalize this behavior, we can see that the high resistive state combinations are easier to distinguish than the low resistive ones.

In general, writing to a resistive memory cell is demanding more resources in terms of energy and delay than executing a read or a neuromorphic inference operation. This gives an incentive to reduce the number of writing operations during the March test sequence. Please note that in the normal March sequence for conventional CMOS memories, there is no difference in read and write operation on test time. However, for testing neuromorphic hardware based on resistive memories, the objective of test time minimization can be translated into reducing the number of write operations. Therefore, we try to activate the faulty behavior which is otherwise activated by specific operand combinations (i.e., requiring write operation in the test sequence) by shifting their respective operation reference with the help of reference trimming and thus changing the effective sense margin. With this concept it is possible to test different defects under the same input combinations without changing the operands. As shown in Figure 1 we have evaluated three operations with shifted references. The first is a lower MIN^41 operation (MIN1L), which is set in the middle of the original R_{MIN1} and the resistance of all cells in LRS. The second and third are lower and higher versions of the MIN^44 operation (MIN4L/MIN4H), which are offset by the original MIN^41 sense margin SM1, as indicated by the SM1 arrows.

B. Fault analysis results based on reduced sense margins

We have performed extensive defect injection and fault analysis for different threshold functions (MIN^Nm) with normal and reduced sense margins. Figure 2 shows the faults with the highest sensitivity with regards to the different MIN^4m operations under test for each defect. The top four graphs depict the most restricting test patterns for open faults. Therefore, if a circuit has to be able to perform all MIN operations, the most dominated open defect which corresponds to lowest possible resistive defect has to be tested. In case of e.g. an open on the bit-line [see the top left image in Figure 2 - (1)BL] the smallest possible defect to lead to a fault is sensitized by the MIN4 operation with an operand input of '0111'. Likewise, minimum resistive defects can be found for opens on the word-line, source-line and the internal node.

The remaining graphs show the results of the evaluated short defects. Here the most dominated fault results from the largest resistive defect (ideally, the resistance between two unconnected wires should be infinite). So by testing for the operation with the highest resistance of short defects, all operations with lower defect resistances are covered.

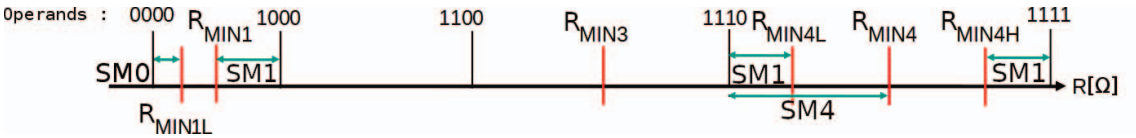

Fig. 1: Relative resistive states of four enabled Cells for in-memory computation and the relevant reference resistances for the evaluated threshold operations. The operands encode a LRS with 0 and a HRS with 1.

IV. Neuromorphic Test Sequence Generation

Based on the neuromorphic fault analysis results presented in the previous section, here we derive different test sequences to detect all single faults for all threshold functions. We start by looking at the problem from pure logic testing perspective and then consider the technology-specific mapping based on resistive memories. Finally, we take advantage of reference trimming to derive the optimal test time.

A. Full Logic Test

Let $MIN^N m$ be the threshold operation to evaluate if at least m bit out of N ($m \leq N$) are set. From a pure logic testing perspective of all possible threshold(MIN) operations, each threshold function $MIN^N m$ requires two sets of test inputs. The first set of test inputs generate the input combination at the exact threshold value (with m 1's and $N-m$ 0s) to detect all (single) stuck-at-0 faults on the output as well as the inputs set to 1. The second set of test patterns correspond to the input combinations just below the threshold (with $m-1$ 1s and $N-m+1$ 0s) to detect all (single) stuck-at-1 faults, on the output as well as the inputs set to 0. However, as the upper threshold test of one operation is the lower threshold test of the next threshold, half of the test patterns can be shared with testing the previous threshold function MIN^N*m-1*.

Lemma 4.1: Given a memory with N rows, the test complexity C_{full} for the logic test for all $MIN^N m$ operations with 100% single stuck-at coverage is 2^N.

Proof: Testing each $MIN^N m$ operation requires $C(N,m) + C(N,m-1)$ test patterns, in which $C(n,k) = n!/k!(n-k)!$. Therefore, to test all MIN operations the total number of test patterns is $\sum_{i=1}^{N} C(N,i) = 2^N$. This shows that pure logic testing of a neuromorphic array implementing all possible threshold functions requires all possible combinations of memory values with an exponential complexity.

B. Minimal-Maximal Threshold Test

While the pure logic testing has exponential complexity, the fact that threshold functions are implemented by resistive memories can help to reduce the test complexity. One key observation of the fault injection results in Figure 2 is that all defective behaviors can be detected by only checking the two extreme threshold functions, $MIN^N 1$ and $MIN^N N$. This is because of the resistive behavior of the cell and the sensing mechanism to implement threshold functions, which allows us to reduce the test complexity significantly compared to the pure logic test. To confirm this, we characterized the defective behavior of $MIN^4 1$, $MIN^4 3$ and $MIN^4 4$ operations experimentally and evaluated the trend between $MIN^4 1$ and $MIN^4 4$. We can see from the results, that the minimum test set can be generated by only using the operations $MIN^4$1(0000), $MIN^4$1(0100), $MIN^4$1(1000) and $MIN^4$4(0111).

From this we can generate a generic non-marching test sequence as follows. By interpreting the four-bit operations as N-bit threshold operations we can formulate MIN^N1(0000) as $MIN^N 1$ on all zero-cells. Therefore, we start by writing the entire array to zero and perform a single $MIN^N 1$ operation which expects a 0 as the operation result. $MIN^4$1(0100) and $MIN^4$1(1000) can be interpreted as a "running 1" test, which performs the $MIN^N 1$ operation after each step. This is not the usual behavior of marching tests, as the neuromorphic operation is performed in lockstep with the write operation. We formulate this as a loop, which uses the current cell address as an index, sets the cell at the previous address to 0, the next cell to test to 1 and then performs the $MIN^N 1$ operation which expects a 1 as a result. $MIN^4$4(0111) can consequently be interpreted as a "running 0" and a $MIN^N N$ operation performed after each step. We therefore write the entire array to 1 and perform the same steps as above with adjusted writing operations and $MIN^N N$ which expects a result of 1.

Assuming that a "running 1/0" write will take 2N write operations (toggle two operands per step), the entire test will have a complexity of 8N+1 (plus one single $MIN^N 1$ operation). More specifically, the write complexity is 6N while performing 2N+1 neuromorphic read (inference) operations.

C. Minimal Write Test using Trimming

Typically, to mitigate process variation of reference resistors and sense amplifiers in conventional STT-MRAM, trimming circuits [16] are used to increase/decrease the effective resistance of reference resistors dynamically after manufacturing. It can be implemented by a series of resistors, which are bypassed individually depending on a control signal. This allows the reading circuit to be configured during the test phase to minimize the read failures.

In the previous section, we showed that the threshold operations are most sensitive in their minimum/maximum threshold operation ($MIN^N 1$ and $MIN^N N$). With this in mind, we propose to use resistance trimming to mimic the sense margins of other operations. For example, as shown in Figure 1 by decreasing R_{MIN4} to R_{MIN4L}, we can emulate a sense margin for $MIN^4 4$ operations and operands, which are based on the sensitivity of the $MIN^4 1$ operation.

By further decreasing the minimum sense margin, which is usually available with trimming circuitry, it is possible to activate particular defects under certain threshold operation and operand values, which are otherwise only detected with different threshold operations under certain operands. This

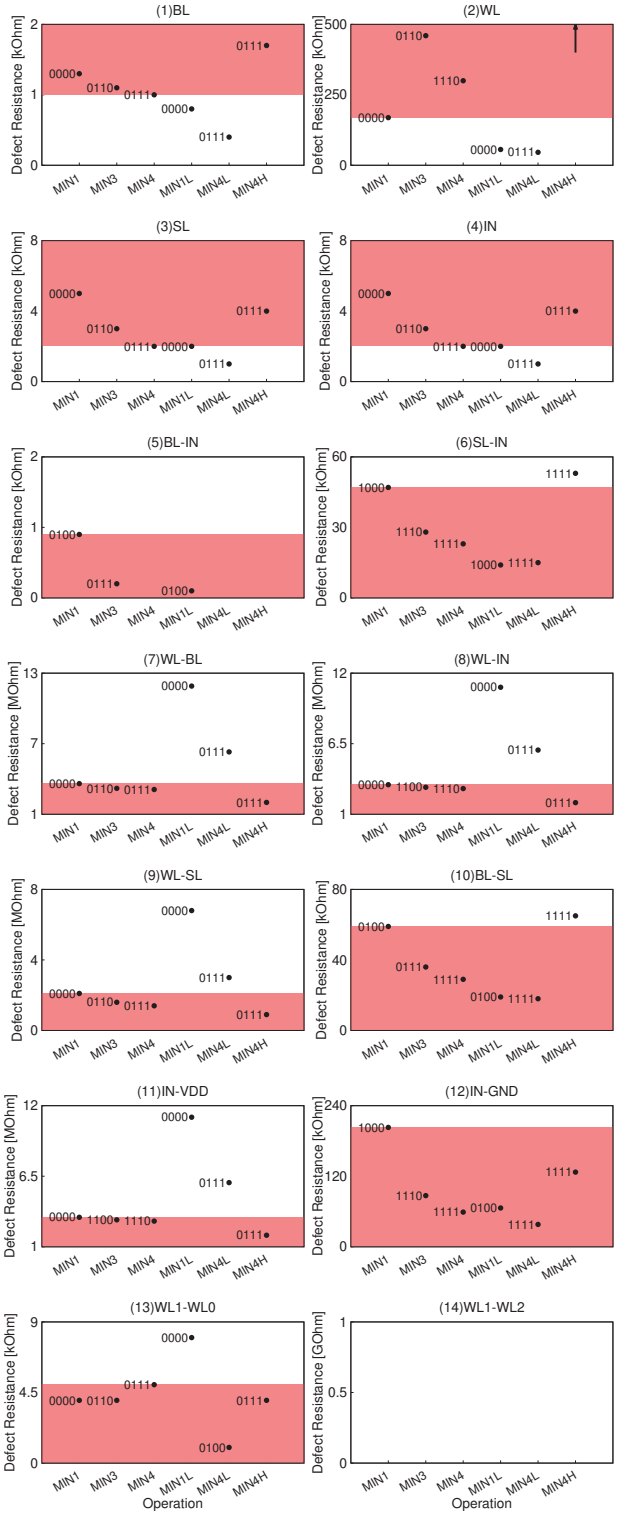

Fig. 2: Fault injection results for the four-bit operations: Each injected fault is shown with the most restricting input configuration (depicted next to the respective point) for each evaluated threshold operation. A defect in the red marked area results in at least one faulty behavior during the standard MIN1-MIN4 operations. MIN4 operations with a bridging defect between BL and IN (5) and no operation with a bridging defect between WL1 and WL2 (14) resulted in an observable fault.

allows us to avoid writing sequences to create other operand values and instead change the sense margin with the same inputs to detect faults. For this purpose, we can sensitize more faults with the MIN^N1 operation by reducing the sense margin to half, as indicated by SM0 in Figure 1, and eliminate the need for MIN^NN operation and other operand values. The minimum test set can thus be generated with only $MIN^41L(0000)$, $MIN^41(0100)$ and $MIN^41(1000)$.

A full generic test derived the same way as in the previous subsection is therefore as follows. We start by writing the whole memory to zero and perform an MIN^N1L operation expecting zero as a result. Then we step through the memory, performing one write-0 to the previous tested cell and a write-1 to the current tested cell and perform a MIN^N1 operation expecting a result of one. The test complexity for this algorithm is 4N, with 3N write operations and N+1 neuromorphic read operations. This test is therefore able to reduce the test complexity by another 50% compared to our previous maximal threshold test.

V. Conclusion

In this paper we investigated the testing of neuromorphic arrays based on resistive memories. We particularly used reference resistance trimming to vary the sense margin of the threshold operations to activate faults only detectable under different threshold operations and operands, to further reduce the test time. With this approach we generated a test sequence, which is able to detect all single faults in all threshold operations used in neuromorphic-enabled resistive memories.

References

[1] H.-. P. Wong *et al.*, "Metal–oxide rram," *Proceedings of the IEEE*, vol. 100, no. 6, 2012.

[2] ——, "Phase change memory," *Proceedings of the IEEE*, vol. 98, no. 12, 2010.

[3] A. D. Kent and D. C. Worledge, "A new spin on magnetic memories," *Nature nanotechnology*, vol. 10, no. 3, 2015.

[4] S. Jain, A. Ranjan, K. Roy, and A. Raghunathan, "Computing in Memory With Spin-Transfer Torque Magnetic RAM," *TVLSI*, no. 3, 2018.

[5] S. M. Nair, C. Münch, and M. B. Tahoori, "Defect characterization and test generation for spintronic-based compute-in-memory," in *IEEE European Test Symposium (ETS)*, 2020.

[6] S. M. Nair *et al.*, "Defect Injection, Fault Modeling and Test Algorithm Generation Methodology for STT-MRAM," in *ITC*, 2018.

[7] C.-Y. Chen *et al.*, "RRAM defect modeling and failure analysis based on march test and a novel squeeze-search scheme," *IEEE Transactions on Computers*, vol. 64, no. 1, 2014.

[8] S. Kannan *et al.*, "Sneak-path testing of crossbar-based nonvolatile random access memories," *IEEE Trans. Nanotechnol.*, 2013.

[9] T.-L. Tsai, J.-F. Li, C.-L. Hsu, and C.-T. Sun, "Testing of In-Memory-Computing 8T SRAMs," in *DFT*, 2019.

[10] M. Fieback, S. Nagarajan, M. Taouil, and S. Hamdioui, "Testing Computation-in-Memory Circuits," in *ITC*, 2019.

[11] A. Emara *et al.*, "Testing of memristor ratioed logic (MRL) XOR gate," in *ICM*, 2016.

[12] A. Gebregiorgis and M. B. Tahoori, "Testing of neuromorphic circuits: Structural vs functional," in *ITC*, 2019.

[13] C. Münch and M. B. Tahoori, "Defect characterization of spintronic-based neuromorphic circuits," in *IOLTS*, 2020.

[14] S. Hamdioui, M. Fieback *et al.*, "Testing computation-in-memory architectures based on emerging memories," in *ITC*, 2019.

[15] A. Mejdoubi *et al.*, "A compact model of precessional spin-transfer switching for MTJ with a perpendicular polarizer," in *MIEL*, 2012.

[16] A. Antonyan, S. Pyo, H. Jung, and T. Song, "Embedded mram macro for eflash replacement," in *ISCAS*, 2018.