Preliminary Defect Analysis of 8T SRAM Cells for In-Memory Computing Architectures

Lila Ammoura, Marie-Lise Flottes, Patrick Girard, Arnaud Virazel

Preliminary Defect Analysis of 8T SRAM Cells for In-Memory Computing Architectures

L. Ammoura M.L. Flottes P. Girard A. Virazel

LIRMM – University of Montpellier, CNRS
Montpellier, France
<lastname>@lirmm.fr

***Abstract*—In-Memory-Computing (IMC) paradigm has been proposed as an alternative to overcome the memory wall faced by conventional von Neumann computing architectures. IMC architectures proposed today are built either from volatile or non-volatile basic memory cells, but a common feature is that all of them are prone to manufacturing defects in the same way as conventional memories. In this paper, we propose to analyze the behavior of an IMC 8T SRAM cell in presence of defects located in the read port of the cell. A model of a basic IMC memory array has been set up to simulate the behavior of the cell in the two modes of operation: memory mode and computing mode. Resistive short defects were injected into the read port and then analyzed. Preliminary results show that these defects can severely impact the behavior of the 8T SRAM in memory mode as well as computing mode. The final goal of this study is to develop effective test algorithms for these defects.**

Keywords—8T SRAM cell, In-Memory Computing, test, defective cell, resistive defect, memory mode, computing mode.

I. INTRODUCTION

Most modern computer systems are based on the von Neumann architecture, which is characterized by memory storage decoupled from the processing cores. However, the efficiency of data-intensive applications such as artificial intelligence, cryptography, search engines, etc. is now severely impacted by the von Neumann bottleneck [1]. This problem is caused by frequent and large data transfers between memory units and processing cores, resulting in high energy consumption and overall throughput limitation [1]. To overcome these performance and power penalties, many efforts have been done to develop new architectural paradigms.

One of the most promising alternatives is to use In-Memory Computing (IMC) architectures [2][3]. Beyond their classical storage function, these memory architectures aim at integrating extra logic in the memory array and in the periphery to circumvent the von Neumann bottleneck problem. By this way, computations and operations can be performed directly inside the memory.

IMC architectures can be built by using various types of volatile or non-volatile basic memory cells. DRAM-based and SRAM-based IMC architectures have been proposed for applications such as graphic accelerators or machine learning [4][5][6]. On the other hand, data-intensive applications can be managed by IMC architectures based on memristive devices, such as Resistive RAM (RRAM), Spin Transfer Torque RAM (STT-MRAM) and even Spin Orbit Torque RAM (SOT-RAM) cells [7][8][9].

A common feature of all these architectures is that they are prone to manufacturing defects in the same way as conventional memories built using the same technologies. Moreover, testing IMC architectures must be done in two different modes of operation: i) during memory mode where the memory function (storage, read and write operations) must be tested and ii) during computing mode where operations allowed by the architecture must be tested. So, testing IMC memories is much more difficult than testing conventional memories [2]. Consequently, developing test solutions for IMC architectures is mandatory to enable a widespread use of this new computing paradigm in modern data processing units.

Several test solutions have been proposed in the last years to screen out manufacturing defects in non-volatile IMC architectures based on RRAM or MRAM cells [10-12]. In [10-11], a systematic approach to develop test solutions has been discussed and applied on RRAM-based IMC architectures. In [12], the authors model and evaluate the impact of various defects (opens and shorts) in the bit-cell of STT-MRAM on an IMC architecture and compare these faults with those of conventional memory faults. With this information, they develop new test algorithms to detect IMC specific faults.

To the best of our knowledge, only two solutions for functional testing of IMC memories based on volatile 8T SRAM cells have been reported in the literature [2][3]. In [2], authors propose a March-like test for an IMC memory with SRAM and TCAM functions. The March-like test can fully cover simple functional faults in SRAM mode and comparison faults in TCAM mode. In [3], authors propose a March C−8 test algorithm to cover typical functional faults and process variation-induced faults of an 8T SRAM-based IMC. Although the above solutions allow to functionally test an IMC architecture in both memory and computing modes, none of them has considered the detection of potential defects in the read port of the IMC memory, which is the main part of an IMC that differs from conventional 6T SRAM architectures.

Electrical defect characterization is a preliminary step to identify testability conditions of a defective cell. In [13], the authors analyzed the faulty behavior of an 8T SRAM cell caused by open defects (full and resistive) affecting the terminal nodes of the read port transistors. However, this was done under the assumption that the cell works correctly with respect to the write operation and that only the read operation can cause an erroneous read.

In this paper, we propose an analysis of the IMC 8T SRAM cell w.r.t all resistive short-circuit defects that may occur into the read port of the cell. Defects on the remaining part (six transistors) behave in the same manner as defects in the conventional 6T SRAM cell, and hence have been widely investigated in the literature [14]. From the results of this analysis, our goal is to develop new test algorithms to detect defects specific to IMC memories. These algorithms will be used in conjunction with conventional SRAM test algorithms

to achieve complete fault coverage in 8T SRAM IMC architectures.

The rest of the paper is organized as follows. Section II details the basic operations of an IMC memory and presents the model considered to analyze its behavior. In Section III, we describe our defect injection framework and analyze the results obtained on SPICE simulations using a 28 nm technology. Section IV discusses the algorithms proposed in the literature to test an 8T SRAM IMC memory. Finally, Section V concludes the paper and gives future perspectives.

II. SRAM-based In-Memory Computing

IMC architectures allow computations to be performed directly in the memory instead of offloading the data to an external computing node. They can operate in two modes: memory mode and computing mode. In memory mode, the memory performs a read or write operation on an addressed word. In computing mode, the memory executes a calculation from at least two addressed words.

To characterize the electrical behavior of the 8T SRAM cell in both modes of operation, the model shown in Fig. 1, which consists of two 8T SRAM cells, was considered.

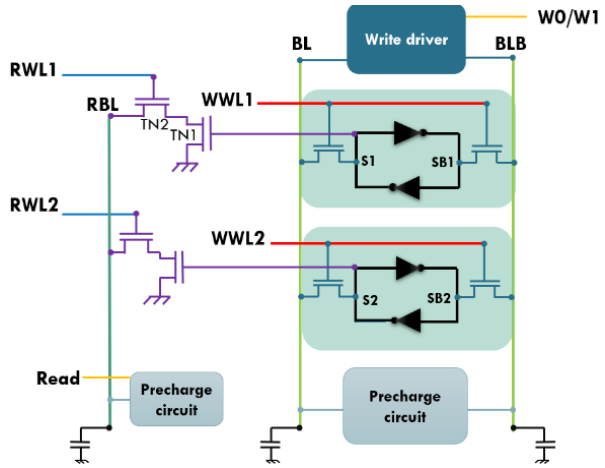

Fig. 1. Considered IMC model

In memory mode, the writing operation in an 8T SRAM cell is similar to that of the 6T. The data is loaded on the Bit Line (BL) and its inverse on the Bit Line Bar (BLB). Then, the concerned Write Word Line (WWL1 or WWL2) is activated. To read the content of cell, the Read Bit Line (RBL), initially precharged at Vdd, remains floating at '1'. Then, the concerned Read Word Line (RWL1 or RWL2) is activated. Let us consider that the first cell stores a '0'. In this case, the NMOS transistor TN1 of the read port whose gate is connected to the memory node S1 will be in the off state. The RBL will therefore be maintained at Vdd, so a '0' will be read on the read output port (considering the presence of an inverter at the "Read" output of RBL). In case the cell stores a '1' (S1 = '1'), on the other hand, the RBL will be discharged through TN1 and TN2, so that a '1' will be read on the read output port after inversion.

In computing mode, the read operation is performed on the two 8T SRAM cells by simultaneously activating their RWLs. The output of the read port (RBL) subsequently shows a NOR behavior of the selected 8T SRAM cells.

III. Defect Analysis of 8T SRAM-Based IMC

This section presents a defect analysis of the IMC 8T SRAM cell. All resistive short-circuit defects that may occur into the read port of the cell have been analyzed. Defects on the remaining six transistors behave in the same manner as defects in the conventional 6T cell and hence have not been considered.

Beforehand, we summarize in the next subsection the results reported in [13], in which open defects in the read port of a 8T SRAM cell have been analyzed.

A. Resistive open and short defect injection

Fig. 2 shows the resistive defects we have injected into the read port of the 8T SRAM cell. Six defects (three open and three short defects) were considered for each transistor.

The effects of three full and resistive open defects (referred as df7, df8 and df11 in Fig. 2 affecting the terminal nodes of the read transistors of the 8T SRAM cell were presented in [13]. During electrical characterization, the authors considered that the values written inside the cell are supposed to be correct and that only the reading operation may cause an erroneous behavior. However, this is not valid in the case of short resistive defects as illustrated in the next sub-section.

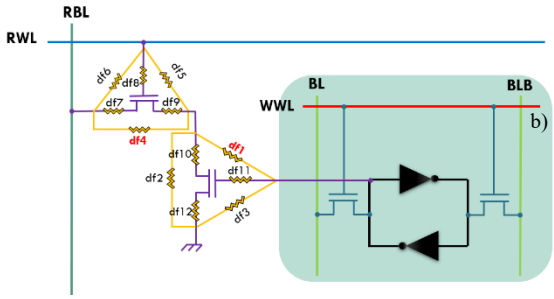

Fig. 2. Resistive open and short defects injected in the read port of an 8T SRAM cell

In [13], the authors also showed that the erroneous effect of an open resistive defect on the read operation on a cell depends not only on the resistance of the defect but may also depend on the previous write operations performed on the cell. Thereby, the sequence of write operations on the faulty cell prior to the read operation may hide the presence of the defect.

In the sequel, resistive short defects df1 and df4 are analyzed. The behavior of the 8T SRAM is simulated each time in the presence of one of the two defects. Note that in our study, we assume that the values preliminary written inside the cell are always correct as defects df1 and df4 only affect the read port.

In the following figures, read operations RX and write operation WX are in green when correctly executed and red when affected by the defect under study. The read operation is checked by the behavior of the RBL, and the write operation by the behavior of the storage node S.

B. Analysis of resistive short defect df1

To analyze the behavior of a defective 8T SRAM cell in presence of df1, let us consider the case where the cell is initially loaded by a '0' (S=0). By activating the RWL, node S will cause the RBL voltage to drop due to defect df1, thus leading to an incorrect R0 operation on the cell.

The behavior of the cell in presence of df1 with a defect resistance of 1Ω (i.e., a pure short defect) is illustrated in Fig. 3. The R0 operation is indeed incorrect since, at t=5ns after the RWL activation, RBL is completely discharged. At t=10ns, the W1 operation is not executed since the voltage at node S has remained at a value lower than Vdd/2.

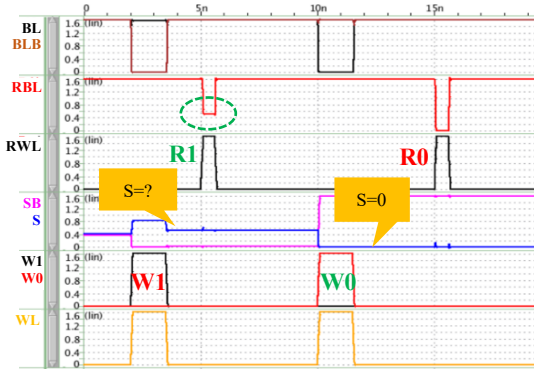

Fig. 3. Simulation of the behavior of a faulty SRAM 8T cell in presence of df1=1Ω

Fig. 4 shows the execution of operations (W1, R1) through the behavior of the storage node S and the voltage of the RBL when the resistance of df1 varies. According to the behavior of the storage node S and for any resistance value lower than 150KΩ, the W1 operation is not performed. Although, the R1 operation is always correct since for any value of the resistance of df1, the voltage at the RBL remains lower than Vdd/2.

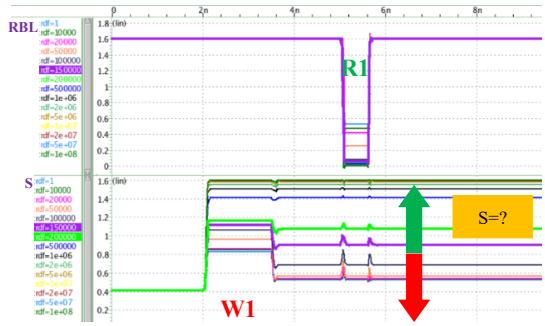

Fig. 4. W1 and R1 operation behaviors depending on the resistance value of defect df1

Fig. 5 depicts the faulty behavior of the RBL during execution of the R0 operation when the resistance of df1 varies. As can be seen, for any resistance value of defect df1 lower than 5 MΩ, the R0 operation is incorrect.

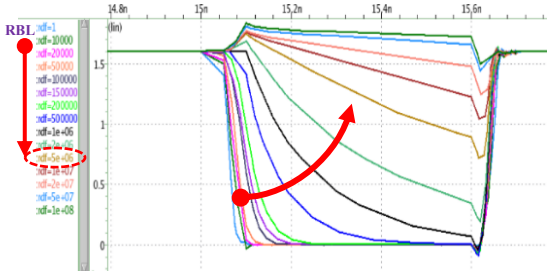

Fig. 5. Execution of the R0 operation depending on the resistance value of defect df1

To summarize, df1 affects the 8T SRAM cell in both memory and computing modes. In memory mode, R0 and W1 operations are corrupted. In computing mode, the NOR(0, 0) function is incorrect too. These faulty behaviors are observed for a defect size lower than 150kΩ.

C. Comprehensive results of resistive defect injections

From the analysis of short and open resistive defects injected in the read port of the 8T SRAM cell, Tables I and II summarize the various faulty behaviors by specifying the resistance range for which defects are detectable.

Table I summarizes the behavior of resistive short defects. Different orders of critical resistance were found, about hundreds of kΩ for df1 and df3, twenties of kΩ for df5 and df6, and in the order of MΩ for df2. These defects can be correctly tested by classical March test algorithms except df4 which is a particular defect affecting all cells of the same RBL.

TABLE I. SUMMARY OF THE FAULTY BEHAVIOR CAUSED BY RESISTIVE SHORT DEFECTS WITH THE RESISTANCE RANGE FOR WHICH THEY ARE DETECTABLE

Resistive short defects	Memory mode	Computing mode	Detectable resistance
df1	R0 incorrect W1 incorrect	NOR (0,0) incorrect	R<150kΩ
df2	R0 incorrect	NOR (0,0) incorrect	R≤5MΩ
df3	W1 incorrect R1 incorrect	NOR (1,0) incorrect	R<150kΩ
df4	Correct behavior	Correct behavior	--
df5	R1 incorrect	NOR (1,0) incorrect	R<15kΩ
df6	R0 incorrect R1 incorrect	Incorrect behavior	R<20kΩ

Table II summarizes the behavior of resistive open defects. Regarding the critical values of these defects, they are of the order of MΩ. The electrical characterization presented in [13] is a preliminary step to identify the testability conditions of such type of defects.

TABLE II. SUMMARY OF THE FAULTY BEHAVIOR CAUSED BY RESISTIVE OPEN DEFECTS WITH THE RESISTANCE RANGE FOR WHICH THEY ARE DETECTABLE

Resistive open defects	Memory mode	Computing mode	Detectable resistance
df7,8,9,10,12	R1 incorrect	NOR (1,0) incorrect	df7,9,10,12 > 5MΩ df8 > 25MΩ
df11	R0 incorrect	NOR (0,0) incorrect	df11 > 10MΩ

IV. DISCUSSION ON TEST SOLUTIONS FOR READ PORT DEFECTS IN 8T SRAM

An IMC memory needs to be tested in its two operating modes, i.e., memory mode and computing mode. Test in memory mode is the same as the one carried out for a conventional memory since the structure of the storage part of the cell of an IMC memory is the same as the one used in a conventional memory.

To detect potential defects, Functional Fault Models (FFM) of a conventional SRAM memory can be used for testing the IMC in memory mode. March tests are widely used

to detect FFMs in SRAMs due to their linear complexity with respect to the memory size. A March test represents a finite sequence of March elements. Each March element contains a finite number of read/write operations that are executed on each cell according to a pre-specified address sequence, which can be traversed in ascending (⇑), descending (⇓), or indifferent (⇳) order [14]. The March C- test: {⇳ (w0); ⇑ (r0, w1); ⇑ (r1, w0); ⇓ (r0, w1); ⇓ (r1, w0); ⇳ (r0)} is a widely used test algorithm for testing SRAMs, with a complexity of 10N (N being the number of cells). This algorithm provides a 100% coverage of Stuck-at-Faults (SAF), Transition Faults (TF), idempotent and inversion Coupling Faults (CFid, CFin) and Address decoder Faults (AF).

The test method proposed in [3] for an 8T SRAM-based IMC with NAND, NOR and XOR operations imposes two requirements:

- Requirement Rq1 consists in performing a read operation or NOR operation to detect the fault caused by leakage current when the data is all 0.
- Requirement Rq2 consists in performing a NAND operation in the worst cases, i.e., when the operands are (0,1) or (1,0).

To satisfy these two requirements, two NAND operations were added to the March C- test algorithm to finally cover the functional faults of the 8T SRAM-based IMC in both operating modes. Modifications have been made to the March C- test algorithm [2] in order to test IMC memory based on configurable 8T SRAM cells in both SRAM and TCAM operating modes.

In summary, [2] and [3] proposed March-like test algorithms to test the proper IMC operation of 8T SRAM memories. However, these March algorithms do not cover all defects that may affect the read port.

From the results of the above analysis about resistive defects injected in the read port, our future work will consist in developing a test algorithm to detect faults of IMC SRAM 8T memories in their two operating modes.

V. Conclusion

In this paper, we first explained the working principle of 8T SRAM cells. Then, we discussed the IMC memories based on 8T SRAMs and we presented the model considered to simulate and analyze their behavior in presence of defects injected in the read port of the cell. Finally, we have discussed existing algorithms proposed so far in the literature to test an 8T SRAM IMC memory in the two modes of operation. Existing test algorithms do not detect all resistive defects in a 8T SRAM cell, especially those located in the read port. The next step in our work will be to establish the testability conditions of an 8T SRAM IMC and develop test algorithms to be used in both modes of operation.

References

[1] A. Agrawal, A. Jaiswal, C. Lee, and K. Roy, "X-SRAM: Enabling In-Memory Boolean Computations in CMOS Static Random Access Memories," *IEEE Trans. on Circuits and Systems I*, vol. 65, no. 12, pp. 4219-4232, 2018.

[2] J.-F. Li, T.-L. Tsai, C.-L. Hsu, and C.-T. Sun, "Testing of Configurable 8T SRAMs for In-Memory Computing," *in Proc. IEEE Asian Test Symposium*, 2020.

[3] T.L. Tsai, J.F. Li, C.-L. Hsu, and C.-T. Sun, "Testing of In-Memory-Computing 8T SRAMs," *in Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, 2019.

[4] J. Zhang, Z. Wang, and N. Verma, "In-memory computation of a machine-learning classifier in a standard 6T SRAM array," *IEEE Jour. of Solid-State Circuits*, vol. 52, no. 4, pp. 915–924, Apr. 2017.

[5] S. Angizi and D. Fan, "GraphiDe: A Graph Processing Accelerator leveraging In-DRAM-Computing," *ACM Great Lakes Symposium on VLSI*, Washington, D.C. USA, May 9-11, 2019.

[6] https://www.eejournal.com/article/in-memory-computing/, In-Memory Computing, No Fewer than Four Approaches, by Bryon Moyer, 2019.

[7] F. Parveen, S. Angizi, and D. Fan, "IMFlexCom: Energy Efficient In-memory Flexible Computing using Dual-mode SOT-MRAM," *ACM Journal on Emgerging Technologies in Computing Systems*, Vol. 14, no.3, Oct. 2018.

[8] Z. He, Y. Zhang, S. Angizi, B. Gong and D. Fan, "Exploring A SOT-MRAM based In-Memory Computing for Data Processing," *IEEE Transactions on Multi-Scale Computing Systems*, 2018.

[9] M. Kooli, H.-P. Charles, C. Touzet, B. Giraud, and J.-P. Noel, "Smart instruction codes for in-memory computing architectures compatible with standard SRAM interfaces," *In Proc. Design, Automation & Test in Europe Conference & Exhibition*, 2018.

[10] M. Fieback, S. Nagarajan, R. Bishnoi, M.B. Tahoori, M. Taouil, and S. Hamdioui, "Testing scouting logic-based computation-in-memory architectures," *In Proc. European Test Symp*, May 2020.

[11] S. Hamdioui, M. Fieback, S. Nagarajan, and M. Taouil, "Testing Computation-in-Memory Architectures Based on Emerging Memories," *In Proc. IEEE International Test Conference*, 2019.

[12] S.M. Nair, C. Munch, and M.B. Tahoori, "Defect Characterization and Test Generation for Spintronic-based Compute-In-Memory," *In Proc. IEEE European Test Symposium*, 2020.

[13] R. Rodriguez-Montanés, D. Arumi, S. Manich, J. Figueras, S. Di Carlo, P. Prinetto, and A. Scionti, "Defective Behaviour of an 8T SRAM Cell with Open Defects," *in Proc. Int. conference on Advances in System Testing and Validation Lifecycle*, 2010.

[14] A. Bosio, L. Dilillo, P. Girard, S. Paravossoudovitch, and A. Virazel, "Advanced Test Methods for SRAMs," ISBN 978-1-4419-0938-1, Springer, 2009.