Analysis of Resistive-Open Defects on a Foundry 8T SRAM-based IMC Architecture

Lila Ammoura, Marie-Lise Flottes, Patrick Girard, Jean-Philippe Noël, Arnaud Virazel

Analysis of Resistive-Open Defects on a Foundry 8T SRAM-based IMC Architecture

L. Ammoura [1] M.-L. Flottes [1] P. Girard [1] J.-P. Noel [2] A. Virazel [1]

[1] LIRMM – Univ. of Montpellier / CNRS
F-34392 Montpellier
<name>@lirmm.fr

[2] Univ. Grenoble Alpes, CEA, LIST
F-38000 Grenoble
jean-philippe.noel@cea.fr

***Abstract*— The adoption of In-Memory Computing (IMC) architectures is one of the promising approaches to efficiently solve the Von Neumann bottleneck problem. In addition to arithmetic operations, IMC architectures aim at integrating additional logic operations directly in the memory array or/and at the periphery for saving time and power consumption. In this paper, a comprehensive model of a 128x128 bitcell array based on a 28nm FD-SOI process technology has been considered to analyze the behavior of IMC 8T SRAM bitcells in the presence of resistive-open defects injected in the read port. A hierarchical analysis including a detailed study of each defect was performed in order to determine their impact both in memory and computing modes, both locally on the defective bitcell and globally on the array. Experimental results show that the IMC mode offers the most effective detectability of resistive-open defects.**

Keywords*— *In-Memory Computing, 8T SRAM cell, resistive-open defect, Test.

I. Introduction

In-Memory-Computing (IMC) paradigm has been proposed as an alternative to overcome the memory wall faced by conventional Von Neumann computing architectures. In addition to arithmetic operations, IMC architectures aim at integrating additional logic operations directly in the memory array or/and at the periphery for saving time and power consumption [1-3].

Defect analysis and, potentially, dedicated test solutions is however mandatory in order to allow a large deployment of these new computing paradigms and related architectures. Two solutions have been proposed in [4-5] for testing correct operations of an 8T SRAM-based IMC architectures in computing mode. These solutions mainly consist in adding computing operations to the original March test algorithm. As shown in [6], these tests however do not cover all potential defects in the targeted IMC architecture. In particular, defects in the memory read port are not covered.

This paper presents a study on resistive-open defects located in the read port of an 8T SRAM bitcell designed using a 28nm FD-SOI process technology. The study has been carried out in order to determine the impact of each of these intra-cell resistive-open defects in both memory (Read/Write operations) and computing modes, locally, on the defective bitcell as well as, globally, on other cells of the array. Impact on sensitization test sequences is also discussed. Experiments show that the operation in IMC mode improves the detectability of resistive-open defects and allows the detection of smaller size defects when considering all the cells of the same column for the computing operation.

II. Resistive-Open Defect Injection Framework

Performing a computation in memory is ultimately equivalent to performing a Read operation on at least two bitcells of the same column. So, ensuring that the read operation operates correctly is essential for any IMC architectures. 8T SRAM bitcells are the most suitable for SRAM-based IMC because they have a read port isolated from the write port. It ensures that the read operation does not interfere with the data content of the bitcell, even if several *RWLs* are activated simultaneously. This behavior makes these cells useful in the IMC context. Therefore, our goal is to analyze the impact of open defects in the read port of 8T SRAM bitcells. Three resistive-open defects are considered for each of the two transistors as shown in Fig. 1.

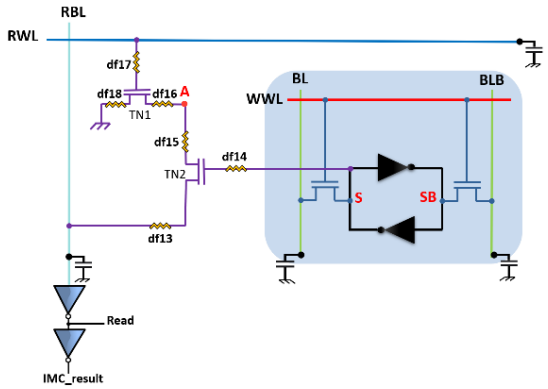

Figure 1. Resistive-open defect injection in the read port of an 8T SRAM cell

To proceed with the injection of resistive-open defects, a monitoring bitcell (*i;j*) (i.e., a bitcell located at row *i* and column *j*) is targeted by a single defect injected at its read port. To analyze each injected defect, we set-up an approach to monitor each time the state of the faulty bitcell (i.e., aggressor cell "c_a"), the states of the neighboring bitcells (i.e., victim cell "c_v" on the same column or row), and the computation results between the faulty bitcell and at least one fault-free bitcell on the same column. The purpose is to reveal the potential impact of each defect on the read/write/computing operations. The defect analysis is hierarchically performed as follows:

- Stand Alone Analysis (SA_Analysis): local impact on the defective bitcell itself during memory mode operations on that bitcell.
- Neighborhood Analysis (N_Analysis): It is done in two steps: i) impact on defect-free surrounding bitcells during memory mode operations on the faulty bitcell, and ii) local impact on the defective bitcell during memory mode operations performed on fault-free surrounding bitcells only.
- Computation Analysis (C_Analysis): It is done in two steps: i) impact on computing mode operations performed between the defective bitcell and at least a fault-free one in the same column i.e., NOR(c_a;c_v), and ii) impact on computing mode operations performed between at least two defect-free bitcells located in the same column than the defective one, i.e., NOR(c_v;c_v).

This hierarchical analysis allows a thorough study of each defect to identify their impact in both memory and computing modes locally on the defective bitcell as well as globally on the array. Moreover, it enables the definition of a Fault Primitive (FP) [7] for each considered defect as follows:

- **<S/F/R>** when a single cell is involved; the cell c_v (victim cell) is used to sensitize a fault where it appears. **S** describes the Sensitizing Operation Sequence (SOS) that sensitizes the fault; **S** ∈ {0, 1, w0, w1, w↑, w↓, r0, r1}.
- **<Sa,Sv/F/R>** when two cells are involved; **Sa** describes the sensitizing operation or state of the aggressor cell, while **Sv** describes those of the victim cell; Si ∈ {0, 1, X, w0, w1, w↑, w↓, r0, r1} (i∈{a, v}), where X is the don't care value X∈{0, 1}.

In both notations**, F** describes the value or the behavior of the faulty cell; **F** ∈ {0, 1, ↑, ↓, -} where ↑ (resp. ↓) means the faulty cell undergoes a transition. **R** describes the logic output level of a read operation in case **S** contains read operations. Generally, it takes one of the values {0, 1, -}, where '-' is used when no read operation is required for the SOS.

III. Experimental Results

A. Experimental Setup

All the electrical simulations of the injected defects have been performed using the XA simulator from Synopsys [8] considering a 128x128 bitcell matrix model designed in 28nm FD-SOI process technology.

All injected resistive-open defects cause a read delay that may induce faulty behaviors. This produced delay depends on the size of the injected defect resistance and the discharge time of the RBL (i.e., parasitic capacitors). Therefore, it is necessary to define a read time T_{read}, at which, we ensure that the data at the output of the read port is captured in the meantime. Thus, T_{read} is deduced from the maximum time T0, which is required by the RBL to completely discharge at the operating environment of a typical process corner, 1V supply voltage and 125°C temperature. Then, T0 is added to a margin of T0/2 as the necessary delay for the data to pass through the read inverters. Based on simulations performed at the operating environment described above, T0 is measured at about 700ps, so T_{read} is selected at 1ns for the rest of the defect injection campaign. To extract the minimum resistance value of each resistive-open defect, a threshold is defined at the level of the RBL voltage which is 30% of Vdd (i.e., 300mv) at the T_{read} instant.

B. Resistive-open defect simulation results

Table 1 summarizes all the results obtained for all the injected resistive-open defects. It details the two categories where the operations affected by the defects appear (SA_Analysis and C_Analysis). The first category (SA_Analysis) is represented in the second part of Table 1 detailing the minimum size of defects that lead to this faulty behavior (i.e., R_{min}). For the category C_Analysis, it is divided into 3 groups (cf. the third the third part of Table 1), the computation is performed between 2 bitcells (i.e., N=1), then 16 bitcells (i.e., N=15) and between all the cells of the column (i.e., N=127), while specifying each time the minimum value of the resistance of the defect which leads to this faulty behavior. For each injected defect, the sequence of operations allowing its sensitization is determined (last column of Table 1) according to the operation that generates the minimum resistance value, i.e., the critical resistance value "$R_c = \min\{R_{min}\}$" of the defect, in order to cover the largest range of these resistive-open defects.

For example, in the case of df13, the minimum defect size is achieved with the C_Analysis with N=127 (i.e., R_c=26.8kΩ) that corresponds to a NOR(1;0^{127}) computing operation. So, the sequence <1,0^{127} NOR(1;0^{127})/0^{127}/1> (detailed below) will be applied considering all the bitcells of the column where the defective bitcell is located as follows:

<1,0^N NOR(1;0^N)/0^N/1>

where a logic '1' is initially stored in the defective bitcell and logic '0' in the N bitcells of the same column as the defective one. Then, a NOR(1;0^N) operation is performed between all the selected bitcells. The N bitcells remains at logic '0'. The output level of the logical operation is a logic '1'.

C. Discussion

According to the defect behavior presented so far for an IMC 8T SRAM bitcell with resistive-open defects at the read port, some detectability conditions can be deduced. As shown in the results reported in Table 1, the IMC mode implying all the bitcells of the same column offers a better detectability of all the injected defects, i.e., an improvement of up to 13.8% for df13 to df16, of 15.7% for df18 and of 1% for df17. Note that the improvement in resistance values is different depending on the location of each defect. Moreover, different ranges of critical resistance have been found. The minimum resistance of defects that are connected to the gates of transistors is in the MΩ range (i.e., df14 and df17), while the

Table 1. Summary of resistive-open defect simulation results

Defect	SA_Analysis		C_Analysis				<S/F/R>/ <Sa,Sv/F/R>
	Operation	R_{min} Ω	IMC Operation NOR(c_a;c_v)	R_{min} Ω N=1	R_{min} Ω N=15	R_{min} Ω N=127	
df13	R1	31k	NOR(1;0^N)	31k	29.8k	26.8k	<1,0^N NOR(1;0^N)/0^N/1>
df14	R1	16.79M	NOR(1;0^N)	16.89M	16.59M	14.56M	<1,0^N NOR(1;0^N)/0^N/1>
df15	R1	23.2k	NOR(1;0^N)	23.2k	22.9k	20k	<1,0^N NOR(1;0^N)/0^N/1>
df16	R1	23.2k	NOR(1;0^N)	23.2k	22.9k	20k	<1,0^N NOR(1;0^N)/0^N/1>
df17	R1	4.39M	NOR(1;0^N)	4.44M	4.42M	4.35M	<1,0^N NOR(1;0^N)/0^N/1>
df18	R1	21k	NOR(1;0^N)	20.6k	20.6k	17.7k	<1,0^N NOR(1;0^N)/0^N/1>

minimum resistance for the other defects (i.e., df13, df15, df16, df18) is in the kΩ range.

From these results, the main conclusion is that the IMC mode improves the detectability of resistive-open defects and allows the detection of smaller size defects by involving all bitcells of the same column for a computing operation. Consequently, the row decoder must be adapted in order to make possible a computing operation involving all bitcells of each column for detecting and covering smaller sizes of resistive-open defects.

IV. Conclusion

In this paper, we presented our analysis for a thorough study of intra-cell resistive-open defects injected into the read port. Impacts in both memory and computation modes were identified, both locally (on the defective bitcell), and globally (on the array). Then, we reported results obtained during the simulation campaigns by specifying the critical size of the defects for which they are detectable. The obtained results show that the IMC mode improves the detectability of the injected resistive-open defects.

Our future work will consist, in a first step, in analyzing the inter-cell resistive-open and resistive-short defects so that, in a second step, it will be possible to develop an effective test and design-for-test solutions that allow to cover all the defects that can affect the IMC 8T SRAM architectures.

References

[1] A. Jaiswal, I. Chakraborty, A. Agrawal and K. Roy, "8T SRAM Cell as a Multibit Dot-Product Engine for Beyond Von Neumann Computing," in IEEE Transactions on Very Large Scale Integration Systems, vol. 27, no. 11, pp. 2556-2567, Nov. 2019.

[2] M. Horowitz, "Computing's energy problem (and what we can do about it)," 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014.

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

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

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

[6] L. Ammoura, M.-L. Flottes, P. Girard and A. Virazel, "Preliminary Defect Analysis of 8T SRAM Cells for In-Memory Computing Architectures," 16th International Conference on Design & Technology of Integrated Systems in Nanoscale Era, 2021, pp. 1-4.

[7] A.J. van de Goor and Z. Al-Ars, "Functional Memory Faults: A Formal Notation and a Taxonomy", VLSI Test Symposium, pp. 281-289, 2000.

[8] "XA User Guide."https://www.synopsys.com/content/dam/synopsys/implementation&signoff/datasheets/primesim-xa-ds.pdf