Analysis of Read Port Short Defects in an 8T SRAM-based IMC Architecture

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

Analysis of Read Port Short Defects in an 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*— A promising new alternative to efficiently solve the von Neumann bottleneck problem is to adopt In-Memory Computing (IMC) architectures. Beyond the arithmetic operations, IMC architectures aim at integrating additional logic operators directly in the memory array or/and at the periphery in order to provide close computing abilities. However, they are subject to manufacturing defects in the same way as conventional memories. In this paper, a comprehensive model of a 128x128 bitcell array based on 28nm FD-SOI process technology has been considered to analyze the behavior of IMC 8T SRAM bitcells in the presence of short defects injected in the read port. A hierarchical analysis allowing a thorough study of each defect has been carried in order to identify their impact in both memory and computing modes, locally on the defective bitcell as well as globally on the array.**

Keywords*— *In-Memory Computing, 8T SRAM cell, short defect analysis, Test.

I. Introduction

Nowadays, dedicated computer systems for data-intensive applications are still based on the Von Neumann paradigm, which introduces major limitations such as reduced performance acceleration, increased power consumption and limited system scalability [1]. A promising approach to efficiently solve the data-intensive applications problem is to adopt In-Memory Computing (IMC) architectures. Beyond conventional operations, IMC architectures aim at integrating additional logic in the memory array and at the periphery in order to provide close computing facilities and efficiently address the Von Neumann bottleneck problem.

In order to enable the use of this new computing paradigm in modern data processing units, the development of test solutions dedicated to IMC architectures is therefore mandatory. Two test solutions have been proposed in [2] and [3] to test the correct operations in computing mode of SRAM-based IMC architectures using 8T SRAM cells. These solutions, mainly consist in modifying March test algorithms through the addition of computing operations. However, as shown by preliminary results presented in [4], these test solutions do not cover all potential defects that can occur in the IMC architecture. In particular, defects in the memory read port are not covered.

In this paper we present a study of short defects in the read port of 8T SRAM bitcells used in IMC architectures. This study has been carried out and validated on an IMC architecture based on 28nm FD-SOI process technology. The goal is to justify the addition of extra computing operations during the execution of the test procedure on an IMC architecture in order to cover all potential short defects in the memory array and IMC operators. The strategy is based on a qualitative study of all potential short defects that may affect the read port of an 8T SRAM bitcell. The study has been carried out to identify the impact of each defect in both memory and computing modes, locally on the defective bitcell as well as globally on the array.

II. IMC SRAM Array

To characterize the electrical behavior under realistic conditions, our study was conducted on the model presented in Fig. 1, which is a 128x128 bitcell array designed in 28 nm FD-SOI process technology [5]. The model is made of write drivers ensuring the writing operations in the bitcells and of precharge circuits, which maintain the *RBL* signals at VDD necessary to perform the read operations and the computation at array level. The model also includes a layout extraction of the parasitic capacitances of the main signals (i.e., *BL*/*BLB*, *RBL* and *RWL*) that reinforces the realistic aspect of the model and allows achieving results that closely approximate those that can be achieved in a real circuit.

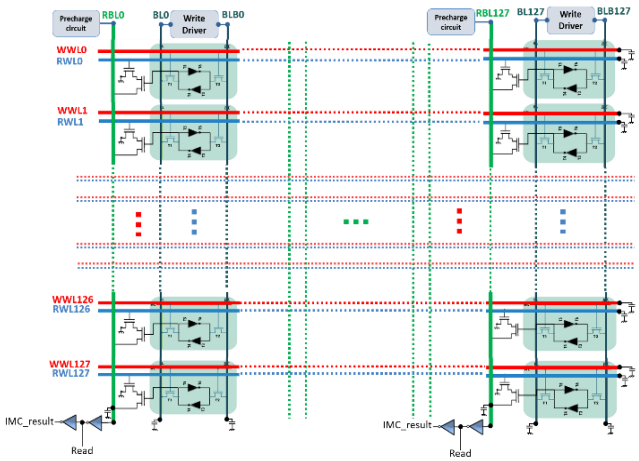

Figure 1. Considered 128x128 matrix model with layout extraction of parasitic capacities

III. Short Defect Analysis

This section first shows the locations of the short defects injected in the read port of the 8T SRAM cell. Then, it details the defect injection approach and finally presents a qualitative analysis of the corresponding faulty behaviors.

A. Short 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. Therefore, our goal is to analyze the impact of short defects in the read port of 8T SRAM cells. Six short defects are considered for each of the two transistors as shown in Fig. 2.

To proceed with the injection of short defects, a monitoring cell (*i; j*) (i.e., 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 cell, the states of the neighboring cells (i.e., cells on the same column and row), and the computation results between a fault-free memory bitcell and the faulty cell. 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 cell.
- Neighborhood Analysis (N_Analysis). It is done in two steps: i) impact on defect-free surrounding cells during memory mode operations on the faulty cell, and ii) local impact on the defective bitcell during memory mode operations performed on fault-free surrounding cells 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, and ii) impact on computing mode operations performed between at least two defect-free cells located in the same column than the defective one.

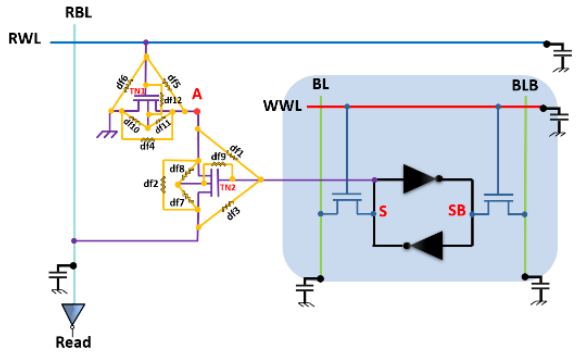

Figure 2. Short defects injection in the read port of an 8T SRAM cell

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) for each considered defect. As detailed in [6], a FP is denoted as:

- **<S/F/R>** when a single bitcell is involved; the bitcell Cv (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 bitcells are involved; **Sa** describes the sensitizing operation or state of the aggressor bitcell, 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 bitcell; **F** ∈ {0, 1, ↑, ↓, -} where ↑ (resp. ↓) means the faulty bitcell 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.

B. *Qualitative Short Defects Analysis*

Faulty behaviors produced by short defects df1 and df4 are described below. We assume the presence of a single defect for each analysis because the occurrence of multiple defects is unlikely.

***Defect Df1*:** This short defect creates a permanent connection between the storage node S and the potential denoted A in Fig. 2.

- SA_Analysis: During a write operation, the *RWL* remains at logic '0', which means that transistor TN1 is always blocked. Consequently, node *S* will not be disturbed during any write operation on the defective bitcell. On the other hand, during a read operation, as soon as the *RWL* is activated, transistor TN1 becomes passing, then node *S* is grounded. Thus, the Read operation is destructive in the case where the defective bitcell initially stores a logic '1'. In other words, R1 operation performed on the defective bitcell operates as a W0 operation and returns a logic '0' to the output port (i.e., inverter of the read port).
- N_Analysis: Since write operations on the defective bitcell are not impacted, any write operation on defect-free surrounding memory cells is correctly acted and vice versa. Faulty behaviors appear during Read operations. Let us consider the case where the faulty bitcell stores a logic '1' and a read operation on a defect-free bitcell of the same row is acted. Upon activation of the *RWL*, the content of the defective bitcell will be forced to logic '0'. In summary, when the defective bitcell stores a logic '1', activating the *RWL* signal to act any Read operation on another bitcell of the same row forces its content to a logic '0'.
- C_Analysis: In computing mode, df1 will present the same destructive Read problem. Let us assume a computation between the defective bitcell storing a logic '1' and a defect-free bitcell on the same column storing a logic '0'. As soon as the appropriate *RWL* signals are activated, the content of the defective bitcell switches to a logic '0'. Consequently, instead of computing the NOR (1;0) operation that must provide a logic '0', the output IMC_result of the read port provides a logic '1' corresponding to the NOR (0;0) (see Fig. 1).

All the injected short defects have been analyzed in the same way as detailed for df1, df4 and df6. Table 1 summarizes the results of the analysis and reports at each analysis step (SA_Analysis, N_Analysis, C_Analysis) the operations affected by each injected defect ("RX"/"WX" when the Read/Write operation is affected, "-" if no operation is affected). For each injected defect, the sequence of operations allowing its sensitization is determined according to the first analysis category in which the first operation impacted by the defect appears (see last column of Table 1). For example, in the case of injection of df1, the first affected operation appears in the category SA_Analysis, so the sequence <1R1/0/0> (detailed below) will be applied on the defective bitcell since the defect is detectable directly on it:

<**1R1/0/0**>: A logic '1' is initially stored in the defective bitcell. Then a R1 operation is acted. The defective bitcell

flips and remains at logic ‘0’. The output level of a Read operation is a logic ‘0’.

Table 1. Summary of the qualitative short defect analysis

Defect	SA_Analysis	N_Analysis		C_Analysis		<S/F/R>/ <Sa,Sv/F/R>
		Same Row	Same Column	IMC with the defective cell	IMC on defect-free cells	
df1	Destructive R1	Destructive Rx	-	NOR(1;0)	-	<1R1/0/0>
df2	R0	-	-	NOR(0;0)	-	<0R0/0/1>
df3	W0, R1	-	-	NOR(1;0)	-	<1R1/1/0>
df4	-	-	R0	-	NOR(0;0)	<1,0R0/0/1>
df5	R1	-	R0	NOR(1;0)	NOR(0;0)	<1R1/1/0>
df6	R1	R1	-	NOR(1;0)	NOR(0;0)	<1R1/1/0>
df7	R0	-	R0	NOR(0;0)	-	<0R0/0/1>
df8	-	-	R0	-	NOR(0;0)	<X,0R0/0/1>
df9	W1, R1	-	-	NOR(1;0)	-	<1R1/0/0>
df10	-	-	-	-	-	-
df11	-	-	R0	-	NOR(0;0)	<X,0R0/0/1>
df12	R1	R1	-	NOR(1;0)	-	<1R1/1/0>

IV. Experimental Results

In this section, we present the simulation results for short defects df1 analyzed in the previous subsection. A summary of the observed faulty behaviors of the twelve injected defects is provided at the end.

A. Short Defect Simulation Results - Df1 Analysis

Waveforms in Fig. 3 present the faulty behavior of the memory in presence of short defect df1 with typical PVT conditions (Process Typ, Voltage 1V and Temperature 27°C).

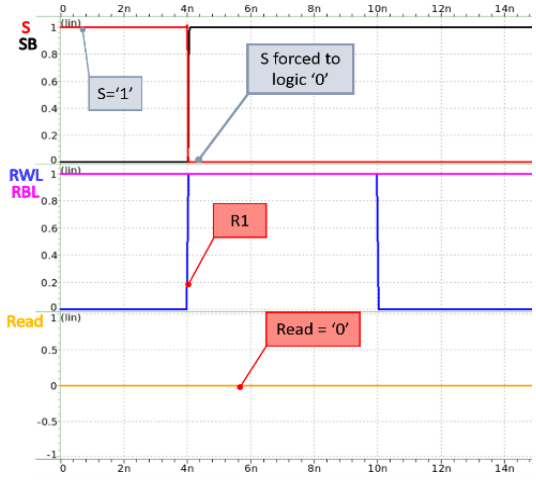

Figure 3. Waveforms of <1R1/0/0> Simulation (df1)

The simulation is performed on a defective bitcell in presence of the short defect df1, and initially contains a logic ‘1’. We directly apply a R1 operation, which produces a W0 operation by forcing the storage node *S* to ‘0’ and returns at the read port output a logic ‘0’ (i.e., Read signal in Fig. 3). So, in the case of df1, the R1 operation is destructive because the content of the bitcell switches and returns a logic ‘0’ at the output of the read port.

B. Discussion on the Results of Short Defects Injection

The qualitative analysis obtained in the previous section was validated through simulations, by applying the sensitization sequences of each defect on the real considered architecture.

Short defects df1, df2, df3, df5, df6, df7, df9 and df12 are detectable by simple memory operations (Read/Write) applied on the defective bitcell. The three short defects df4, df8 and df11 are detectable by memory operations applied on a victim bitcell (i.e., defect-free bitcell) of the same column as the defective bitcell. Conversely, df10 is not detectable because it does not affect the operation in either memory or computing mode.

From these results, the first conclusion to be drawn is therefore: all the injected short defects can be sensitized by simple memory operations (Read/Write), applied either on the defective bitcell or on a victim bitcell. In other words, the results obtained show that to cover the short defects of the read port, a March-type test algorithm is sufficient without the addition of computing operations that would require more energy and more time to be applied on a memory array.

V. Conclusion and Perspectives

In this paper, we first presented the realistic memory model considered in our study (128x128 bitcell array in 28nm FD-SOI process technology). Then, we presented our applied hierarchical analysis for a thorough study of each short defect injected into the read port. Impacts in both memory and computation modes were identified, both locally (on the defective cell), and globally (on the array). Then, we extracted the sensitization sequences and we validated them by simulation on the considered architecture. The results obtained show that all the injected short defects can be detected by a simple March-type test algorithm without having to introduce extra energy-consuming computing operations.

In our future work, impact of each defect for different resistance values will be investigated.

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

[3] 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.

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

[5] N. Planes et al., “28nm FDSOI technology platform for high-speed low-voltage digital applications,” 2012 Symposium on VLSI Technology, 2012.

[6] 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.