A Design-for-Diagnosis Technique for SRAM Write Drivers

Alexandre Ney, Patrick Girard, Christian Landrault, Serge Pravossoudovitch, Arnaud Virazel, Magali Bastian Hage-Hassan, Vincent Gouin

▶ **To cite this version:**

Alexandre Ney, Patrick Girard, Christian Landrault, Serge Pravossoudovitch, Arnaud Virazel, et al.. A Design-for-Diagnosis Technique for SRAM Write Drivers. DATE: Design, Automation and Test in Europe, Mar 2008, Munich, Germany. pp.1480-1485, 10.1109/DATE.2008.4484883 . lirmm-00341796

HAL Id: lirmm-00341796
https://hal-lirmm.ccsd.cnrs.fr/lirmm-00341796v1

Submitted on 6 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Design-for-Diagnosis Technique for SRAM Write Drivers

A. Ney [1] P. Girard [1] S. Pravossoudovitch [1] A. Virazel [1] M. Bastian [2] V. Gouin [2]

[1] *Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier – LIRMM*
Université de Montpellier II / CNRS
161, rue Ada – 34392 Montpellier Cedex 5, France
Email: {ney, girard, pravo, virazel}@lirmm.fr URL: http://www.lirmm.fr/~w3mic

[2] *Infineon Technologies France*
2600, route des Crêtes – 06560 Sophia-Antipolis, France
Email: {magali.bastian, vicent.gouin}@infineon.com URL: http://www.infineon.com

Abstract*

Diagnosis is becoming a major concern with the rapid development of semiconductor memories. It provides information about the location of manufacturing defects in the memory, and its effectiveness allows a fast yield ramp up. Most of existing diagnosis methods uses a fault dictionary to provide detailed information of fault localization. However, these solutions are most of the time unable to distinguish between all faults, and more importantly often fail to identify the actual faulty block of the memory. Identifying which block of a memory (core-cell array, write drivers, address decoders, pre-charge circuits, etc ...) is defective allows saving considerable amount of time during the ramp up phase.

In this paper, we propose a very low cost Design-for-Diagnosis (DfD) solution for identifying faulty write drivers. It consists in verifying logic and analog conditions that guarantee the fault-free behavior of the write driver. The proposed solution allows a fast diagnosis (only three consecutive write operations are needed to fully diagnose the write driver) and induces a low area overhead (about 0.5% for a 512x512 SRAM). Beside diagnosis, an additional interest of such a solution is its usefulness during a post-silicon characterization process, where it can be used to extract the main features of write drivers (logic and analog levels on bit lines).

1. Introduction

Embedded memories occupy a major portion of the silicon area and consume most of the transistors of a typical SoC [1]. Therefore, the yield of such system mainly depends on the yield of the embedded memories. Test and diagnosis of embedded memories are thus two important challenges in SoC development.

Test methods for SRAMs are generally based on March algorithms [2]. These March tests target static faults such as stuck-at, transition and coupling faults. Recently, dynamic faults [3, 4, 5], which are faults requiring more than one operation in sequence to be detected, have been considered. New March tests, such as March RAW [6] targeting dynamic faults in core-cells, or modifications of existing March tests, such as March iC- [7] targeting dynamic faults in address decoders, were proposed.

As soon as the application of March algorithms has revealed logic errors in a given memory, diagnosis can be performed. Generally, diagnosis approaches are based on the cause-effect paradigm as for logic design diagnosis [8]. To measure the quality of a diagnosis algorithm, the diagnosability ratio (DR) is used. It is defined as the ratio of the number of distinguishable fault types among the number of total detectable fault types. Diagnosis methods generally use a fault dictionary and try to achieve the highest diagnosability ratio for a given test algorithm [9, 10, 11, 12]. However, these solutions are most of the time unable to distinguish between all faults (or all fault models) and hence do not allow to determine which block of the memory is defective. In fact, a fault model may be related to more than one defect in the various blocks of the memory. For example, it has been shown in [13, 14] that a transition fault can be due to defects locating in a core-cell or in a write driver. In this case, a diagnosis solution based on a fault dictionary will indicate that the memory is affected by a transition fault without any information on the faulty block of the memory.

The overall objective of this study is to provide a mean to identify which block of a memory (core-cell array, write drivers, address decoders, pre-charge circuits, etc …) is defective. Even if around 80% of the silicon area of a memory is taken by the core-cell array, which is hence

* This work has been funded by the French government under the framework of the MEDEA+ 2A702 "NanoTEST" European program.

978-3-9810801-3-1/DATE08 © 2008 EDAA

more prone to defects than any other block, providing such type of information can save considerable amount of time during the ramp up phase in case of a malfunction coming from outside the core-cell array.

A first step in this work is to diagnose faulty write drivers by proposing a very low cost DfD solution. The next step will consist in providing similar diagnosis solutions for other blocks of the memory.

A typical write driver operation consists in pulling-down one of the two bit lines of a given cell depending on the write operation type (w0, w1) while maintaining the other bit line at Vdd. The resulting voltage levels on bit lines during a write operation are a good indicator of the write driver correctness and reliability. The proposed DfD solution allows verifying the voltage levels on both bit lines during write operations. This operation can be done during the diagnosis phase and can easily determine the defective write driver (if any) in the memory. This solution allows saving a considerable amount of time during the ramp up phase in case of a malfunction coming from a defect in one of the write drivers. Moreover, the proposed diagnosis solution is also able to determine weak write drivers. A weak write driver is a driver that may act a write operation correctly even if it does not drive the correct voltage levels on both bit lines. Although such a weak driver will not be identified during the test phase, it may impact the reliability of the whole memory as a malfunction can manifest after a certain amount of time during the lifetime of the memory. We show that the proposed DfD solution must be designed depending on voltage levels on bit lines that have to be measured and requires an activation sequence of only three write operations. The proposed hardware solution has a low impact on the area overhead (about 0.5% for a 512x512 SRAM). Beside diagnosis, an additional interest of such a solution is its usefulness during a post-silicon characterization process, where it can be used to extract the main features of the write drivers (logic and analog levels on bit lines). This type of information is particularly useful for high safety applications, such as automotive or medical applications.

This paper is organized as follows. Section 2 describes a typical write driver structure and its functioning. Section 3 discusses the conditions needed to guarantee the fault-free behavior of the write driver. Then, Section 4 presents and validates the proposed DfD solution. Finally, concluding remarks and future works are discussed in Section 5.

2. Write driver fault-free operation

By groups of columns in an SRAM, a write driver is used to control the true bit line (BL) and the complement bit line (BLB) during a write operation. As the two bit lines are pre-charged at Vdd before every operation, the write driver has just to perform the pull down of one of the two bit lines during a write operation:

- BL for a write '0' (w0) operation
- BLB for a write '1' (w1) operation

In our study, we consider the write driver structure presented in Figure 1. It is composed of a write control part and a driver part. The first part receives the data that has to be written (DataIn) and the Write Enable signal (active at low level) which controls the write operation with its two outputs, named AW0 and AW1. If DataIn = 0 and the write enable signal is active, then AW0 = 1 and AW1 = 0. In this case, transistor Mtn1 acts the pull down of BL which corresponds to a w0 operation. In the same way, if DataIn = 1 (w1 operation), AW0 = 0 and AW1 = 1, so that transistor Mtn2 performs the pull down of BLB. At this point, it is important to notice that for a fault-free write driver, signals AW0 and AW1 can never be set to 1 at the same time.

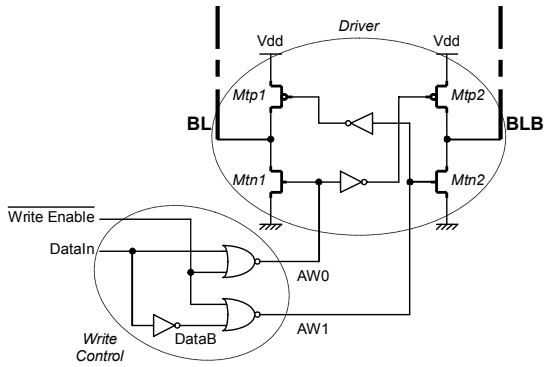

Figure 1: Write driver structure

Waveforms presented in Figure 2 show the correct action of the write driver during two consecutive write operations. Especially, we perform a w1 followed by a w0 on a cell that initially contains a '0'. In Figure 2, S and SB are the state values of the accessed core-cell. These waveforms were obtained for typical operating conditions, *i.e.* process: typical, voltage: 1.2V, temperature: 27°C.

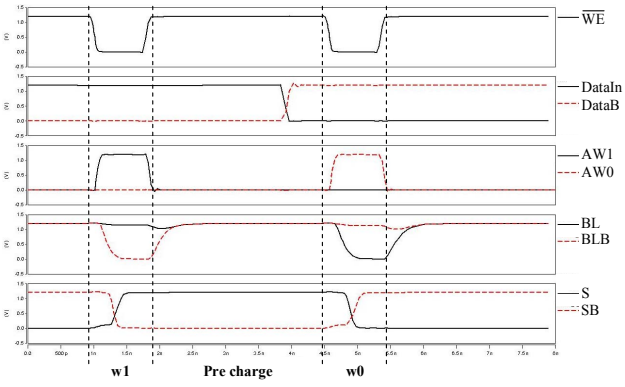

Figure 2: Fault-free write driver waveforms (w1, w0)

3. Requirements for fault-free operation of a write driver

In the previous section, we have presented the fault-free operation of the SRAM write driver. Now, we can enumerate the two important conditions that are needed to guarantee the fault-free behavior of the write driver – a logic and an analog conditions.

3.1. Logic condition

As shown in Section 2, the write driver must act the pull down of one of the two bit lines. The other bit line is maintained at Vdd during the write operation. From this statement, we can extract a first condition for a fault-free operation of the write driver:

$$BL \oplus BLB = 1 \qquad \text{(Eq. 1)}$$

If this equation is not satisfied during a write operation, then it means that both bit lines present the same voltage level. In case of Vdd, no write operation is performed. Conversely, the two bit lines at Gnd indicate that both w0 and w1 operations are performed simultaneously.

This first condition allows performing a logical diagnosis of the write driver. Nevertheless, it does not allow verifying the exact voltage level driven on the bit lines during the write operation. Thus, an additional analog condition is needed to diagnose weak write drivers.

3.2. Analog condition

Voltage levels on bit lines during write operations are a major concern when embedded memories are used for high safety applications. In fact, over the lifetime of a product, memories are exposed to many phenomena which degrade their performances. For this reason, it is important to verify the good voltage level of bit lines after manufacturing. A wrong level at this early stage of the lifetime of the memory indicates a weakness of the write driver, which can be degraded over the time and lead to erroneous write operations. So, in addition to the logic condition, an analog condition has to be satisfied to guarantee the good voltage levels on the bit lines.

The write driver can be seen as a current source that has to discharge one bit line and to maintain the other at Vdd. During a fault-free operation (w0) let us consider that it delivers a current I_{ideall} for the discharge of bit line BL and I_{idealh} for bit line BLB. Thus, a weak write driver delivers less current than I_{ideall} (resp. I_{idealh}). Consequently, at the end of the write operation, the level of the bit line that has to be discharged is higher than Gnd (resp. the level of the bit line that has to be maintained at Vdd is less than Vdd). This can be view on waveforms in Figure 3 where a w0 operation is performed by a fault-free write driver (top of Figure 3) and a weak write driver (bottom of Figure 3).

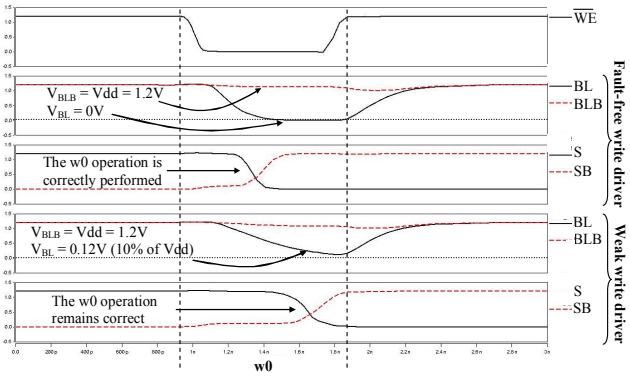

Figure 3: Fault-free and weak write driver operations

From this statement, we can extract two analog conditions for a fault-free operation in case of a w0 operation. Note that the analog conditions for a w1 operation can be derived in the same way.

$$I_{reall} \geq \alpha^1 \bullet I_{ideall} \text{ with } 0 \leq \alpha^1 \leq 1 \qquad \text{(Eq. 2.a)}$$
$$\Rightarrow V_{BL} \leq \beta^1 \bullet Vdd \text{ with } 0 \leq \beta^1 \leq 1$$

and

$$I_{realh} \geq \alpha^2 \bullet I_{idealh} \text{ with } 0 \leq \alpha^2 \leq 1 \qquad \text{(Eq. 2.b)}$$
$$\Rightarrow V_{BLB} \geq \beta^2 \bullet Vdd \text{ with } 0 \leq \beta^2 \leq 1$$

where α^1 and α^2 represent the strength of the write driver. Parameters β^1 and β^2 are derived form the α parameters and represent the level of charge and discharge of the bit lines. An ideal write driver will be defined by Equations 2.a and 2.b with $\alpha^1 = 1$ and $\alpha^2 = 1$ implying $\beta^1 = 0$ ($V_{BL} = 0V$) and $\beta^2 = 1$ ($V_{BLB} = Vdd$).

Parameters α have to be selected depending on the memory technology and desired reliability level. In our case, we have considered a 65nm SRAM technology and we have chosen parameters as follows:

- α^1 insuring $V_{BL} \leq 0.1 \bullet Vdd$
- α^2 insuring $V_{BLB} \geq 0.7 \bullet Vdd$

Consequently, the write driver will be considered as faulty if it cannot discharged BL at a voltage lower than 10% of Vdd and maintain BLB at a voltage level higher than 70% of Vdd.

These logic and analog conditions can be translated into a design for diagnosis solution for SRAM write drivers as shown in the following section.

4. Description of the DfD solution

The proposed DfD solution consists in adding a hardware module to verify both logic and analog conditions presented in the previous section. Note that we only present how to diagnose a weak or wrong w0

operation. The study of the w1 operation can be derived in a similar way.

4.1. Hardware diagnosis solution for the analog condition

The analog condition consists in verifying if the write driver delivers enough current in the bit lines. For a w0 operation, the bit line (BL) must be discharged at more than $\beta^1 \bullet$ Vdd by the current passing through transistor Mtn1 (see Figure 1). Respectively, BLB must be maintained at $\beta^2 \bullet$ Vdd by the current passing through transistor Mtp2 (see Figure 1). A straightforward solution consists in sensing the resulting voltage levels on bit lines by using logic gates designed to have the required threshold voltage. However, such a solution is unpractical for two raisons; (i) the difficulty to designing gates with very low (0.1V) or very high threshold voltages and (ii) the fact that we must sense two different voltages ($\beta^1 \bullet$ Vdd and $\beta^2 \bullet$ Vdd) on each bit line to diagnose weak or wrong w0 and w1 operations. Consequently, in order to use simple CMOS gates to sense bit line voltage levels, we propose to normalize the pass/fail diagnosis threshold voltage on bit lines at Vdd/2 (instead of 10% and 70% of Vdd). This is done by adding two transistors (Mtptest and Mtntest) producing a resistive divider bridge and hence modulating the bit line voltage levels. This principle is presented in Figure 4.

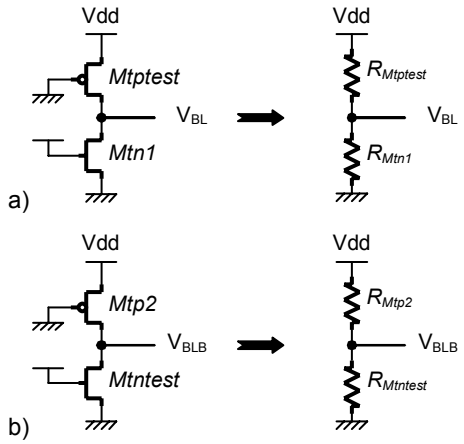

Figure 4: Principle of the DfD solution
a) for the low level and b) for the high level

In a stable state, transistors Mtptest and Mtn1 (resp. Mtntest and Mtp2) can be seen as their equivalent resistances inducing the resistive divider bridge. The strength of Mtptest (resp. Mtntest) is chosen in order to have the following diagnosis conditions:

- if $V_{BL} < \frac{Vdd}{2} \Rightarrow$ the write driver satisfies the analog condition.
- if $V_{BL} > \frac{Vdd}{2} \Rightarrow$ the write driver does not satisfy the analog condition.

To be more precise on the sizing of transistors Mtptest and Mtntest, let us consider Figure 5. It represents I_{DS} as a function of V_{DS} voltage levels of Mtptest and Mtn1 transistors. Note that, in case of Mtntest and Mtp2 transistors, curves are the opposite ones. The stable functioning point corresponds to a sizing of Mtptest insuring less than Vdd/2 on BL. As can be seen on the figure, if the strength of Mtn1 changes (it becomes weak or faulty) the crossing point indicates a resulting voltage level on BL higher than Vdd/2.

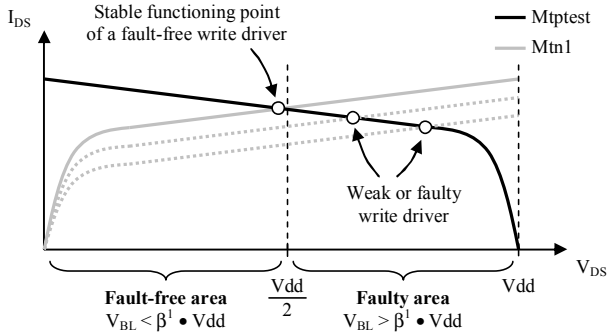

Figure 5: Principle of the diagnosis solution

The hardware implementation of such a principle is presented in Figure 6. It is composed of two parts; the analog structure and the data processing providing the diagnosis result.

The analog structure embeds the two transistors Mtptest and Mtntest plus four transmission gates (MtnpgBL, MtnpgBLB, MtppgBL and MtppgBLB) and two inverters used to isolate and configure the diagnosis module. Two signals ($\overline{W0D}$ and $\overline{W1D}$ active at low level) control the configuration of the analog structure that depends on the write operation type (w0 or w1).

At the end of the write operation, the bit line level reflects the strength of the write driver. The analog structure is designed in order to obtain less than Vdd/2 on BL and more than Vdd/2 on BLB for a fault-free w0 operation. The data processing part allows translating these analog levels into a digital signal. Two inverters are used to amplify the signals and a XOR gate is used to provide the diagnosis results. Node S must be at logic '1' during the write operation in case of a write driver satisfying the analog conditions.

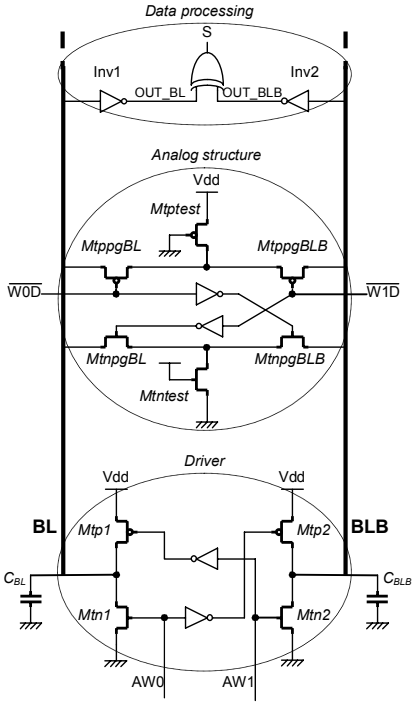

Figure 6: Hardware implementation of the diagnosis module

Waveforms in Figure 7 illustrate the functioning of the proposed structure. Two simulations are superposed; a fault-free write driver simulation (continuous lines) and a weak write driver simulation (dotted lines).

At the beginning of the simulation, BL and BLB are pre-charged at Vdd. Then a w0 operation is performed, leading to AW0 = 1 and AW1 = 0. The diagnosis module is activated with $\overline{W0D}$ = 0 and $\overline{W1D}$ = 1. Then, BL node is discharged and reaches a level lower than Vdd/2 in case of a fault-free write driver. In case of a weak write driver, as transistor Mtn1 has not enough strength to discharge the bit line, V_{BL} remains higher than Vdd/2. As diagnosis result, node S provides a logic '1' in case of a fault-free write driver and a logic '0' for a weak write driver.

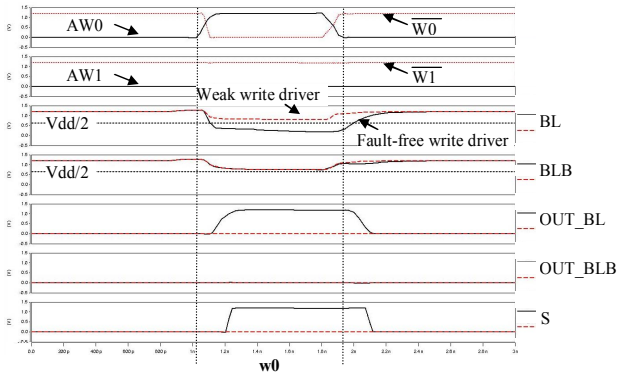

Figure 7: Diagnosis module functioning

Although efficient, such a structure is only able to verify if one bit line has a level lower than Vdd/2 and the other has a level higher than Vdd/2, irrespective of the type of write operations. Additional logic must therefore be added to distinguish between w1 and w0 operations as presented in the next sub-section.

4.2. Hardware diagnosis solution for the logic condition

Based on the previous comment, we must adapt the logic condition (see Eq. 1) so that it can distinguish between w0 and w1 logic levels on bit lines. The solution we propose consists in comparing the bit line logic levels with the data to be written (node DataIn in Figure 1). The new logic condition becomes:

$$\left(\overline{BL} \oplus DataIn\right) \bullet \left(\overline{BLB} \oplus \overline{DataIn}\right) = 1 \qquad \text{(Eq. 3)}$$

It results on some modifications in the initial hardware implementation presented in Figure 6, especially on the data processing part as shown in Figure 8.

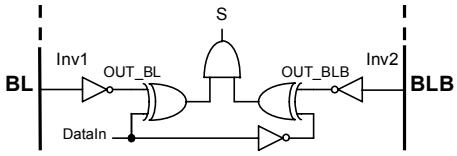

Figure 8: Data processing part of the diagnosis module

Waveforms in Figure 9 show the simulation results of a faulty write driver which is not detectable with the initial data processing module. This faulty write driver always performs w0 operations even if it is configured to perform a w1 operation.

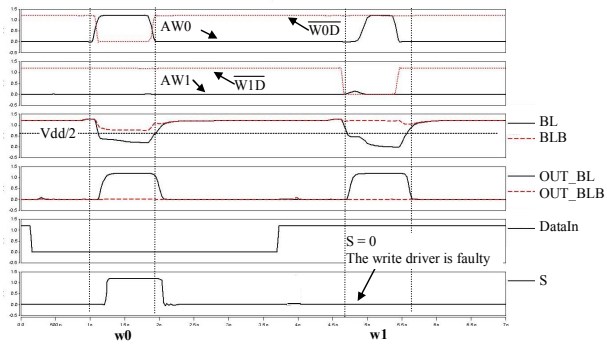

Figure 9: Simulation results of a faulty write driver

In Figure 9, a w0 operation is first performed. BL becomes lower than Vdd/2 while BLB remains higher than Vdd/2. Node S is at logic '1' indicating that the write driver is fault-free. Then, the write driver is configured to perform a w1 operation, *i.e.* node DataIn is set to a logic '1'. As the driver can always perform w0 operations, node AW1 remains at logic '0' while AW0 = 1. As can be seen in Figure 9, BL is lower than Vdd/2 and BLB remains higher than Vdd/2. In such case, the initial data processing part (see Figure 6) would provide a logic '1' on node S indicating a fault-free write driver. With modifications

presented in Figure 8, node S provides a logic '0' that corresponds to a faulty write driver.

The DfD solution is effective and requires about 40 transistors by write driver. For example, it represents 0.5% of area overhead for a 512x512 SRAM.

4.3. Diagnosis sequence

The proposed diagnosis module is able to verify the logic (Eq.3) and analog (Eq. 2a and 2b) conditions. Obviously, a w0 and w1 operations are needed to diagnose the write driver. In this case, only defects involving a static behavior will be diagnosed.

As mentioned in many published studies, defects in VDSM technology may also induce dynamic behaviors. Resulting fault models are dynamic faults [3, 4, 5] as those that may affect the write driver [14]. In this study it is shown that two successive opposite write operations must be performed to detect dynamic fault that may affect the write driver. Consequently, the diagnosis sequence able to deal with static and dynamic faulty behaviors as well is the following:

$$wx\ w\bar{x}\ wx$$

So, only three operations are needed to fully identify a faulty or weak write driver. After applying this diagnosis sequence on the memory, reading the data on node S provides the required information on the correctness of the write drivers.

5. Conclusions and future works

In this paper we have proposed a low cost DfD solution for SRAM write drivers. It allows to identifying wrong or weak write drivers by verifying logic and analog conditions that guarantee the fault-free behavior of the write drivers. Moreover, it allows a fast diagnosis (only three write operations are needed) and induces a low area overhead (about 0.5% for a 512x512 SRAM). An additional interest of such a solution is its usefulness during a post-silicon characterization process, where it can be used to extract the main features of the write drivers.

In a future work, we want to extend this DfD solution to make it able to deal with others blocks connected to the bit lines such as pre-charge circuits and sense amplifiers. The overall objective of this study will be to propose a global DfD solution to identify which block of the memory is defective.

References

[1] Semiconductor Industry Association (SIA), "International Technology Roadmap for Semiconductors (ITRS)", 2005.

[2] A.J. van de Goor, Testing Semiconductor Memories, Theory and Practice, COMTEX Publishing, Gouda, The Netherlands, 1998.

[3] 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.

[4] Z. Al-Ars and A.J. van de Goor, "Static and Dynamic Behavior of Memory Cell Array Opens and Shorts in Embedded DRAMs", Design Automation and Test in Europe, pp. 496-503, 2001.

[5] S. Hamdioui, R. Wadsworth, J.D. Reyes and A.J. van de Goor, "Importance of Dynamic Faults for New SRAM Technologies", European Test Workshop, pp. 29-34, 2003.

[6] S. Hamdioui, Z Al-Ars and A.J. van de Goor, "Testing Static and Dynamic Faults in Random Access Memories", Proc. IEEE VLSI Test Symposium, pp. 395-400, 2002.

[7] L. Dilillo, P. Girard, S. Pravossoudovitch, A. Virazel and S. Borri, "March iC-: An Improved Version of March C- for ADOFs Detection", VLSI Test Symposium, pp. 129-134, 2004.

[8] M. Abramovich et al., "Digital System Testing and Testable Design", IEEE Press, 1990.

[9] M.F. Chang, W.K. Fuchs and J.H. Patel, "Diagnosis and Repair of Memory with Coupling Faults", IEEE Transactions on Computers, vol. 38, no. 4, pp. 493-500, April 1989.

[10] V.N. Yarmolik, Y.V. Klimets, A.J. van de Goor, S.N. Demidenko, "RAM diagnostic tests", Memory Technology, Design and Testing, pp. 100-102, 1996.

[11] D. Niggemeyer, M. Redeker, E.M. Rudnick, "Diagnostic testing of embedded memories based on output tracing", Memory Technology, Design and Testing, pp. 113-118, 2000.

[12] J.-F. Li, K.-L. Cheng, C.-T. Huang and C.-W. Wu, "March-Based RAM Diagnosis Algorithms for Stuck-At and Coupling Faults", International Test Conference, pp. 758-767, 2001.

[13] S. Borri, M. Hage Hassan, P. Girard, C. Landrault, S. Pravossoudovitch and A. Virazel, "Defect-Oriented Dynamic Fault Models for Embedded SRAMs", Proc. of European Test Workshop, pp. 23-27, 2003.

[14] A. Ney, P. Girard, C. Landrault, S. Pravossoudovitch, A. Virazel and M. Bastian, "Slow Write Driver Faults in 65nm Technology SRAM: Analysis and March Test Solution", Design Automation and Test in Europe, 2007.