Slow Write Driver Faults in 65nm SRAM Technology: Analysis and March Test Solution *

A. Ney [1] P. Girard [1] C. Landrault [1] S. Pravossoudovitch [1] A. Virazel [1] M. Bastian [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: <lastname>@lirmm.fr URL: http://www.lirmm.fr/~w3mic

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

Abstract

This paper presents an analysis of the electrical origins of Slow Write Driver Faults (SWDFs) [1] that may affect SRAM write drivers in 65nm technology. This type of fault is the consequence of resistive-open defects in the control part of the write driver. It involves an erroneous write operation when the same write driver performs two successive write operations with opposite data values. In the first part of the paper, we present the SWDF electrical phenomena and their consequences on the SRAM functioning. Next, we show how SWDFs can be sensitized and observed and how a standard March test is able to detect this type of fault.

1. Introduction

Nowadays, more than 50% of the system on Chip (SoC) area is used to embed different kinds of memory as mentioned by the SIA roadmap [2]. This ratio will grow up to 90% in the next few years making memories the main responsible of SoC yield. Consequently, memory testing is becoming a very important step for SoC development.

Test methods for SRAMs are generally based on static fault models (stuck-at, transition, coupling, etc..). However, in very deep submicron technologies (VDSM), a new type of faulty behavior occurs in some particular configurations and is mainly due to resistive-open defects in vias or contacts. This new class of faults is called dynamic faults [3, 4]. These faults require more than one operation in sequence to be sensitized and are most of the time undetectable with classical March test [5].

In our recent studies, we have analyzed dynamic faults in different parts of SRAM memories. In particular, we have studied dynamic faults occurring in address decoders [6, 7] and core-cells [8, 9]. In both cases, we have investigated the physical origins of the dynamic faults and we have proposed efficient March procedures to detect them.

In this paper, we propose an analysis of dynamic faults induced by the presence of resistive-open defects in the write driver of SRAMs using the 65nm Infineon technology. We have inserted resistive-open defects in some locations of a write driver circuit and we have performed electrical simulations in order to evaluate their effects. We have analyzed the functional influence of each single defect on the memory operations. We have demonstrated that defects in the control part of the write driver involve an incorrect write operation when two write operations (with opposite data) are acted sequentially. The fault model associated to the analyzed defects is the Slow Write Driver Fault (SWDF) [1]. Such a fault model requires specific read/write sequence to be sensitized and observed. In the last part of the paper, we show how SWDFs can be sensitized and observed and how a standard March test is able to detect this type of fault.

This paper is organized as follows. Section 2 presents the write driver scheme and functioning. Section 3 summarizes the defect injection process and the faulty behaviors obtained. Section 4 presents in detail the SWDF. In Section 5, we present the required conditions to detect SWDFs and we demonstrate that they can be detected by standard March algorithm as March C-. Finally, concluding remarks are given in Section 6.

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 to Vdd before every operation, the

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

978-3-9810801-2-4/DATE07 © 2007 EDAA

write driver has just to act 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.

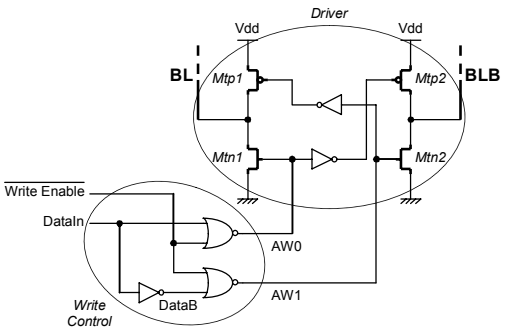

Figure 1: Write driver structure

It is composed by 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 be active, then AW0=1 and AW1=0. In that case, the transistor Mtn1 acts the pull down of BL which corresponds to a w0 operation. In the same way, if DataIn=1, AW0=0 and AW1=1, so that the transistor Mtn2 acts the pull down of BLB. It is a w1 operation. At this point, it is important to notice that, for a non faulty write driver, signals AW0 and AW1 can never be set to 1 at the same time.

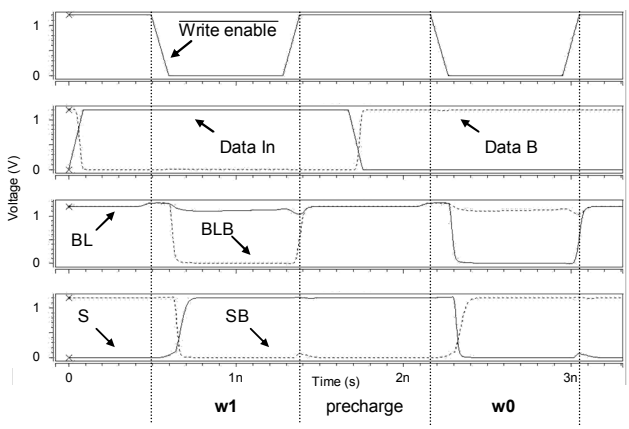

Figure 2: Fault free write driver waveforms (w1, w0)

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'. S and SB are the state values of the core-cell. These waveforms were obtained for typical operating conditions, *i.e.* process: typical, voltage: 1.2V, temperature: 25°C and cycle time: 1.67ns.

3. Resistive-open defects in the write driver

In this section, we analyze the effects induced by resistive-open defects on the normal function of the write driver circuit. We assume the presence of only one defect for each analysis because the occurrence of multiple defects is unlikely.

As shown in Figure 3, nine resistive-open defects (Df1 to Df9) have been placed in different locations of the analyzed circuit. We do not consider all possible locations because of the symmetry of the write driver structure. In particular, we have chosen the left part of the driver for defects Df1 to Df4. Finally, two defects (Df5 and Df6) have been considered in the inverter and three defects (Df7 to Df9) in one of the NOR gates of the write control part. Symmetric defects can be placed on the other NOR gate of the write control part and in the right part of the driver.

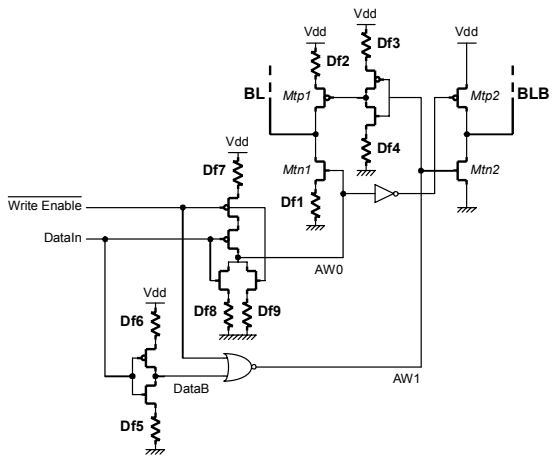

Figure 3: Defect injection in the write driver

Let us now analyze how the different defects in the write driver can disturb the correct operation of the memory. The whole range operating conditions has been considered with the aim of determining the test conditions which maximize the fault detection probability. Hence, simulations have been performed by applying a number of different test patterns with the following varying parameters:

- Process corner: slow, typical, fast, fast n / slow p, slow n / fast p
- Supply voltage: 1.08V, 1.2V, 1.32V
- Temperature: -30°C, 27°C, 110°C
- Defect size has been swept from a few s up to several M s.

Table 1 presents a summary of the fault models identified for each injected resistive defect, along with the conditions for maximum fault detection, *i.e.* the minimum detected resistance value. The definitions of the fault models reported in Table 1 are the following:

- Transition Fault (TF): *A core-cell is said to have a TF if it fails to undergo a transition (0 → 1 or 1 → 0) when there is a write operation.*
- Slow Write Driver Fault (SWDF)*: a write driver is said to have a SWDF if it cannot act a w0 (w1) when this operation is preceded by a w1 (w0). That results on the core-cell that does not change its data content.*

Defect	Process corner	Voltage (V)	Temp (°C)	Min Res (kΩ)	Fault Model
Df1	fast	1.08	- 30	0.4	TF
Df2	-	-	-	-	-
Df3	-	-	-	-	-
Df4	-	-	-	-	-
Df5	fast	1.08	- 30	128	SWDF
Df6	sf	1.32	110	170	SWDF
Df7	fast	1.32	- 30	9.5	TF
Df8	-	-	-	-	-
Df9					(TBD)

Table 1: Summary of worst-case PVT corners for the defects of Figure 3 and corresponding minimum detected resistance and fault model

As shown in Table 1, some defects do not involve a faulty behavior. It is especially the case of defects that control the pull up of node BL (Df2, **Df3** and Df4). In presence of one of these defects, the pull up of node BL cannot be performed but, as the write driver has also a precharge circuit, the pull up is acted any how. In the same way, Df8 prevents the pull down of node AW0 but the pull down is acted by the parallel NMOS transistor control by the write enable signal.

Two defects, Df1 and Df7, involve a transition fault which means that a write (in our case a w0) cannot be performed. Such a fault model can easily be detected by standard March test. For this reason, these two defects will not be considered in the following.

Df9 involves a faulty behavior which is denoted as TBD in Table 1. In presence of this defect, a w0 operation can be performed by the write driver, *i.e.* AW0 node can be set to logic '1'. Normally, at the end of the write operation, the write enable signal acts the pull down of node AW0. However, Df9 prevents this pull down and thus node AW0 remains at logic '1' during a certain time depending on the defect size. As a consequence, the driver continues to act a w0 even if a read operation is performed. This faulty behavior requires a deeper analysis that will be described in a future paper.

In the rest of the paper, we focus only on Df5 and Df6 that involve a Slow Write Driver Fault (SWDF).

4. SWDF: Df5 and Df6 analysis

In presence of defects Df5 and Df6, a Slow Write Driver Fault may occur. During a write operation, one of the two bit lines is driven to '0' and the other one remains at Vdd. However, in presence of Df5 or Df6, this operation cannot be performed, especially when there are two successive write operations with an opposite value.

On this basis, SWDFs can be defined with four FPs (Fault Primitives) [3]. A FP is denoted as **<S/F/R>**. **S** describes the sensitizing operation sequence that sensitizes the fault. **F** describes the value or the behavior of the faulty cell; **F** ∈ {0, 1, ↑, ↓, -}. **R** describes the logic output level of a read operation in case S contains read operations. From this notation, we obtain four FPs for SWDFs, which are divided in two groups. The first group corresponds to defect Df5:

FP1: <1w0w1/0/-> A '1' is initially stored in the cell. Then, a w0 is acted immediately followed by a w1. The cell remains at 0.

FP2: <0w0w1/0/-> A '0' is initially stored on the cell. Then, a w0 is acted immediately followed by a w1. The cell remains at 0.

The second group of FPs corresponds to defect Df6:

FP3: <0w1w0/1/-> A '0' is initially stored on the cell. Then, a w1 is acted immediately followed by a w0. The cell remains at 1.

FP4: <1w1w0/1/-> A '1' is initially stored on the cell. Then, a w1 is acted immediately followed by a w0. The cell remains at 1.

As the data initially stored in the cell does not influence the behavior of the write driver, the following equivalences between FPs can be done:

FP1≡ FP2 and FP3≡ FP4

Consequently, we focus only on FP1 and FP3. Note that, SWDF is a dynamic fault as it requires two consecutive operations (two write operations) to be sensitized.

Waveforms in Figure 4 present the faulty behavior of the memory in presence of Df5 with typical PVT conditions and a defect size of 900 kΩ.

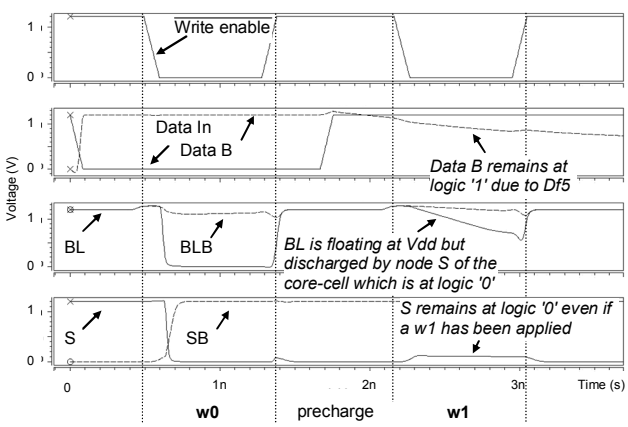

Figure 4: Waveforms of <1w0w1/0/-> simulation (Df5)

Simulation starts on cell that initially contains a '1'. We first apply a w0 operation. Node DataIn is set to '0' and node DataB is set to '1' before the write operation. This first write operation is correctly acted on the cell which

switches from '1' to '0'. Then we act a w1. Just before this operation, DataIn is set to '1' but node DataB remains to a logic '1'. In that case, the pull down of node DataB cannot be performed due to the presence of defect Df5. The two nodes AW0 and AW1 are set to '0'. Any write operation cannot be performed as the four transistors of the drivers (Mtp1, Mtn1, Mtp2 and Mtn2) are off. The two bit lines are floating at Vdd level. This scenario is represented in Figure 5.

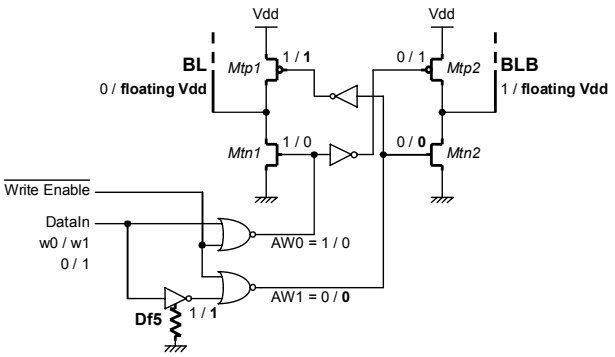

Figure 5: Faulty behavior of the write driver in presence of Df5

As previously, waveforms in Figure 6 present the faulty behavior of the memory in presence of Df6 with the same operating conditions as the ones used for Df5.

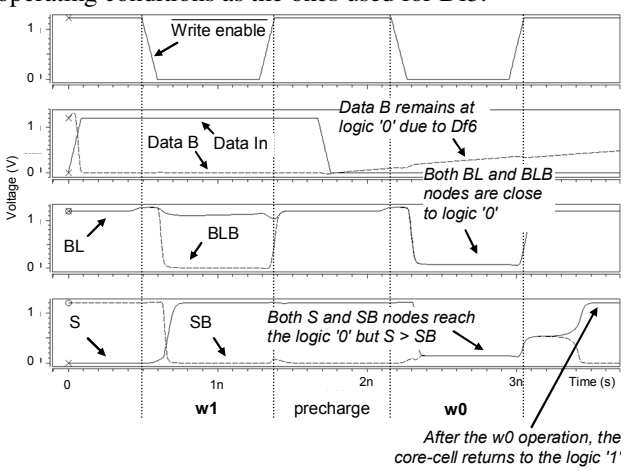

Figure 6: Waveforms of <0w1w0/1/-> simulation (Df6)

This time, the simulation starts on cell that initially contains a '0'. We first apply a w1 operation. Node DataIn is set to '1' and node DataB is set to '0' before the write operation. This first write operation is correctly acted and the cell switches from '0' to '1'. Then, we act a w0. Just before this operation, DataIn is set to '0' but node DataB remains to a logic '0'. In that case, the pull up of node DataB cannot be performed due to the presence of defect Df6. The two nodes AW0 and AW1 are at logic level '1'. This configuration is problematic as it means that the driver must act simultaneously a w0 (AW0=1) and w1 (AW1=1). From an electrical level point of view, the four transistors of the driver are on. Thus, there is resistive short between Vdd and the ground nodes.

In order to define the level of BL and BLB nodes, we must analyze the size but also the purpose of each transistors of the driver. For the same size, it is well known that NMOS transistors are stronger than PMOS transistors. For primitive gates (INV, NAND, NOR etc …), the sizing of N and P plans is done in a way to balance their current driving capabilities. P plans are therefore larger than the N plans. In our case, the problem is different. The driver must act the pull down of one of the two bit lines which are equivalent to non negligible capacitances due to their length. The pull up of the two bit lines is done by the PMOS (Mtp1 and Mtp2) of the driver which is helped by the precharge circuit. However, a specific sizing is done to have the N plan (Mtn1 and Mtn2) at least 5x stronger than the P plan (Mtp1 and Mtp2) and hence insure the pull down of the bit line (BL for a w0 and BLB for a w1) in the time allowed for the write operation. With this specific sizing, the resulting voltages on BL and BLB are then close to '0' during the w0 operation as seen in Figure 6. This level on the two bit lines disturbs the core cell content (nodes S and SB) but after the w0 operation, the core-cell returns to logic '1'. This scenario is represented in Figure 7.

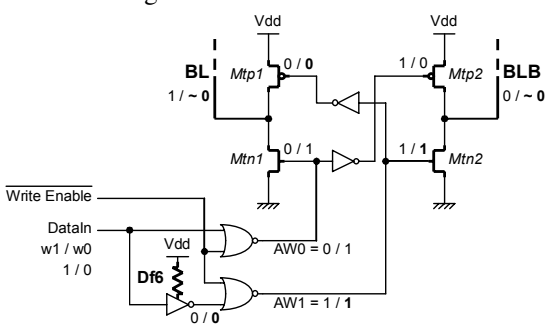

Figure 7: Faulty behavior of the write driver in presence of Df6

The two defects have the same consequences on the memory behavior although the electrical phenomena are a little bit different. The faulty behavior results in a bad write operation if the write is performed after another write with opposite data.

5. March test solution to detect SWDFs

As seen in the previous section, Df5 and Df6 involve SWDFs which is a dynamic fault as it requires two successive write operations to be sensitized. From the FPs presented in the previous section, we can find the required successive operations to detect (sensitize and observe) SWDFs:

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

where the two write operations are for sensitization of the write driver and the read operation is for observation. x = 0 corresponds to the detection of Df5 (Df6). Let us first assume that these three operations must be applied on the same core-cell. From that statement, it is easy to create a

specific March test to detect essentially SWDFs as presented is [1]; March WDm (4N complexity) and March WDw (8N complexity). However, from a test point of view, it is more interesting to obtain a March test that covers not only SWDFs but rather a larger set of fault models. So, we have focused our study on finding possibilities to embed (with additional March elements) or find (with modifications based on the degrees of freedom of March tests [10]) the required succession of operations for SWDFs detection in existing March algorithms.

To do that, we have first to consider again the requirements presented above. Let us assume the basic view of an SRAM array as shown in Figure 8 in which the write driver (WD) is shared by four columns. As the goal is to detect possible malfunction of the WD, it is not necessary to act the three operations on the same core-cell. In fact, the first write operation can be applied on one cell among the cells of the four columns. Then, it is not necessary to act the second write on the same cell but, at least, act this write on a cell of the four columns that initially contains an opposite data to the data used for the write operation. Of course, the read operation has to be performed on the last selected cell to control if the second write operation has been performed correctly. This statement makes the requirements less stringent.

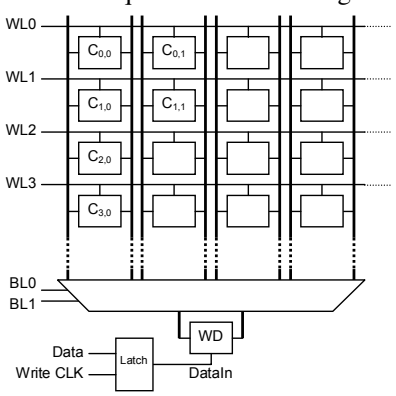

Figure 8: Basic view of a part of an SRAM array

In addition, we can further reduce the stringency of the required conditions to detect SWDFs. This time, we have to look deeper in the write driver structure, especially in the driver control part. It is controlled by a Write Enable signal to perform the write operation with a certain data applied on the DataIn input (see Figure 1). This data is latched, that means, a '0' ('1') is captured in the latch for a w0 (w1) operation. An important property is that when a w0 (w1) is acted by the driver, this data (DataIn) remains stable in the latch as long as another write is not performed. In our case, the latch of the driver captures the first data that has to be written. Thus, it is not necessary to act immediately the second write to sensitize the write driver. Any other operation can be performed between the two write operations as long as it does not use the considered write driver. In the same way, the read operation can be preceded by read or write operations which do not change the content of the faulty core-cell. The resulting successions of operations to detect SWDFs are presented in Figure 9.

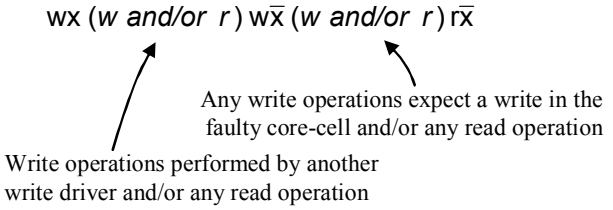

Figure 9: Required conditions to detect SWDFs

From these new and less stringent test conditions, we can try to find them in an existing March test. The March algorithm must have the following requirements:

i. The elements of the March test have to include w0 operations followed by w1 operations to sensitize SWDFs induced by Df5 and w1 operations followed by a w0 operation for those induced by Df6.

ii. The presence of r1 operations is necessary for observation of SWDFs due to Df5 and r0 operations for those induced by Df6.

These two requirements can easily be found in many March algorithms. What is proposed here is to analyze if a well know March algorithm is able to detect SWDFs. In our study, we consider the March C- algorithm, which is often used in industry, here presented in Figure 10.

$$\updownarrow(w0)\ \uparrow(r0, w1)\ \uparrow(r1, w0)\ \downarrow(r0, w1)\ \downarrow(r1, w0)\ \updownarrow(r0)$$
M0 M1 M2 M3 M4 M5

Figure 10: March C- structure

This March algorithm is composed by six March elements and has a 10N complexity. We first consider the succession of M0, M1 and M2 March elements. M0 performs an initialization of the array at logic '0'. During this operation, the DataIn node of each write driver of the structure is latched at '0'. Then, we act element M1 that start by a r0 operation. This operation does not influence the write drivers. The first time we act the w1 operation, the DataIn of the selected write driver is changed from '0' to '1'. This sensitizes the write drivers one after the other in SRAM. Finally, the r1 operation in element M2 performs the observation of possible fault effects. The succession of the three first elements (M0 to M2) allows the detection of SWDFs induced by Df5 (detected by w0w1r1). Table 2 summarizes the actions of elements M0 to M2 on a simple 8 core-cell memory, composed by 2 word lines, 4 bit lines and two write drivers as presented in Figure 11. In order to perform the March elements, we have randomly selected the $\uparrow$ addressing order as follow:

Cell 0, 6, 1, 2, 5, 3, 7, 4

The $\downarrow$ addressing order is of course the reverse one.

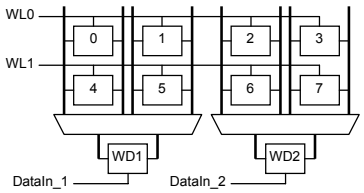

Figure 11: A simple 8 core-cell SRAM memory

Cell N°	Element M0							
0	w0							
1			w0					
4								w0
5					w0			
DataIn_1	0	0	0	0	0	0	0	0
Cell N°								
2				w0				
3						w0		
6		w0						
7							w0	
DataIn_2	x	0	0	0	0	0	0	0

a)

Cell N°	Element M1							
0	r0	w1						
1				Sensitization of WD1				
4								
5								
DataIn_1	0	1	1	1	1	1	1	1
Cell N°								
2				Sensitization of WD2				
3								
6			r0	w1				
7								
DataIn_2	0	0	0	1	1	1	1	1

...

b)

Cell N°	Element M2							
0	r1	w0						
1					r1	w0		
4								
5				Observation				
DataIn_1	1	0	0	0	0	0	0	0
Cell N°								
2							r1	w0
3								
6			r1	w0				
7								
DataIn_2	1	1	1	0	0	0	0	0

...

Table 2: Application of elements M0, M1 and M2 for SWDFs detection

Table 1.a summarizes the action of element M0 on the SRAM depicted in Figure 11. This element acts the initialization of the array at '0'. Then, we perform element M1 (see Table 1.b). First, cell n°0 is read and written to '1'. This w1 sensitizes the first write driver WD1. The same occurs when the w1 operation is performed on cell n°6 which is the first one selected in the second group of columns. SWDFs related to Df5 are thus sensitized. Element M2 (see Table 2.c) performs the observation by acting r1 operations on cell n°0 first (for WD1), and cell n°6 next (for WD2).

In the same way, elements M1, M2 and M3 allow the detection of SWDFs induced by Df6 (detected by w1w0r0). March C- is thus an efficient test algorithm to detect SWDFs in addition to faults (stuck-at, transition, coupling, etc …) initially targeted by this algorithm.

6. Conclusions

In this paper, we have analyzed and characterized the effects of resistive-open defects that may occur in the write driver of SRAMs. We have found that some defects do not disturb the memory behavior, some others involve a transition fault, and two defects (Df5 and Df6) in the inverter of the write control part induce a slow write driver fault.

By performing electrical simulations with the 65nm Infineon technology, we have evaluated the influence of these defects and show that SWDFs can easily be detected by standard March algorithms as March C-.

The next step in this work will be to analyze more precisely the effect of defect Df9 (see Figure 3).

References

[1] A.J. van de Goor, S. Hamdioui and R. Wadsworth, "Detecting Faults in the Peripheral Circuits and an Evaluation of SRAM Tests", Int. Test Conference, pp. 114-123, 2004.

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

[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] Omitted for blind review, "Comparison of Open and Resistive-Open Defect Test Conditions in SRAM Address Decoder", 2003.

[7] Omitted for blind review, "March iC-: An Improved Version of March C- for ADOFs Detection", 2004.

[8] Omitted for blind review, "Dynamic Read Destructive Faults in Embedded SRAMs: Analysis and March Test Solution", 2004.

[9] Omitted for blind review, "Data Retention Fault in SRAM Memories: Analysis and Detection Procedures", 2005.

[10] D. Niggemeyer, M. Redeker and J. Otterstedt, "Integration of Non-classical Faults in Standard March Tests", Records of the Int. Workshop on Memory Technology, Design and Testing, pp. 91-96, 1998.