An Experimental Analysis of Spot Defects in SRAMs: Realistic Fault Models and Tests

Said Hamdioui[1,2] Ad J. van de Goor[2]

[1]Intel Corporation, 2200 Mission College Boulevard, Santa Clara, CA 95052
[2]Delft University of Technology, Faculty of Information Technology and Systems
Section of CARDIT, Mekelweg 4, 2628 CD Delft, The Netherlands
E-mail: said@cardit.et.tudelft.nl

Abstract: *In this paper a complete analysis of spot defects in industrial SRAMs will be presented. All possible defects are simulated, and the resulting electrical faults are transformed into functional fault models. The existence of the usually used theoretical memory fault models will be verified and new ones will be presented. Finally, a new march test detecting all realistic faults, with a test length of 14n, will be introduced, and its fault coverage is compared with other known tests.*

Key words: *SRAMs, fault models, spot defects, march tests, fault coverage.*

1 Introduction

The cost of testing memories increases rapidly with every new generation of memory chips [1]. Precise fault modeling and efficient test design, in order to keep test cost and time within economically acceptable limits, is therefore essential. The quality of the used tests, in terms of their fault coverage and test length, is strongly dependent on the used fault models. Research on current functional tests has shown that the used fault models and tests still leave much to be explained [2].

In [3], *Inductive Fault Analysis (IFA)* and electrical simulation have been applied in order to establish functional fault models for SRAMs using 1.5μ technology; the result was a set of six fault models. In [4], IFA has been applied to SRAMs for yield learning purposes. In [5], a new fault model (i.e, Deceptive Read Disturb Fault) has been introduced based on SPICE simulation, while inserting a resistive defect at the pull-down transistor of the SRAM. In [6], similar work has been done as in [3] using different technologies. The result was a set of fault models which is the same as that reported in [3]; however, with a different probability distribution. In [7], an IFA method has been described, and used in determining the expected yield and test effectiveness. In [8], a structural way of deriving new fault models, based on the insertion of resistive defects into the electrical level of SRAMs, has been reported.

In this paper, a similar method as described in [8] will be used in order to experimentally analyze spot defects in SRAMs. A taxonomy of all possible spot defects in the memory cell array will be given, together with a systematic way of reducing the number of the to-be-simulated defects. SPICE simulation will be performed, and the behavior of the memory cell will be reported by verifying all allowed operations for each defect with a resistance of 0 to $\infty\Omega$. The observed electrical fault behaviors will be transformed into functional fault models. A march test detecting the (re)introduced faults will be presented.

This paper is organized as follows: Section 2 establishes an inventory of spot defects in the SRAM memory cell array. Section 3 presents the experimental results. Section 4 derives the *functional fault models (FFMs)* based on the simulation results. Section 5 represents a test detecting all FFMs of Section 5, and gives an evaluation of the introduced test compared with well-known tests; Section 6 ends with conclusions.

2 Classification of spot defects

Many faults in memory circuits are caused by undesired particles called *spot defects (SDs)*. Depending on their conductivity, they can cause undesired connections or disconnections in the memory. They can be divided into three groups:

Open: an extra resistance within a connection. The resistor value, called R_{op}, is given by $0 < R_{op} \leq \infty$.

Short: an undesired resistive path between a node and V_{cc} or V_{ss}. The resistor value, called R_{sh}, is given by $0 < R_{sh} \leq \infty$.

Bridge: an undesired resistive path between two connections which are not V_{cc} or V_{ss}. The resistor value, called R_{br}, is given by $0 < R_{br} \leq \infty$.

The differential access SRAM memory cell shown in Figure 1 will be considered. All possible opens, shorts and bridges in this memory cell will be defined and located.

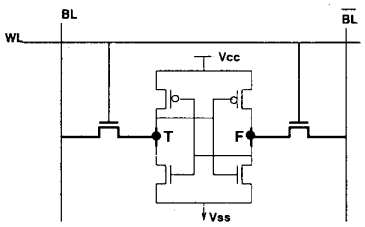

Figure 1. SRAM memory cell

Many defects can be identified in an SRAM. However, due to the symmetrical structure of the cell, only a subset needs to be simulated. For identifying the not-to-be simulated defects, the following terminology is introduced.

- A SD1 shows a **complementary behavior** to SD2 if SD1 and SD2 present defects whose locations in the memory cell are symmetrical with respect to each other. In this case the functional fault behavior of SD1 is similar to that of SD2, with the only difference that all 1's are replaced with 0's and vice versa. E.g., if due to the presence of SD1 the operation read 0 (r0) causes an up transition in the cell, then in the presence of SD2 the r1 operation causes a down-transition in the cell.
- A SD1 (involving two cells) shows an **interchanged behavior** to a SD2 (involving the same two cells) if the fault behavior of SD1 is similar to that of SD2, with the only difference that the aggressor and the victim cells are interchanged.
- A SD1 shows an **interchanged complementary behavior** to SD2 if SD1 shows a complementary and interchanged behavior to SD2.

2.1 Definition and location of opens

Opens in the memory cell can be classified as opens within a cell (denoted as OC), and opens at bit lines (denoted as OB) and at word lines (denoted as OW).

2.1.1 Opens within a cell

In order to define all possible opens, the cell will be considered (without bit lines and word lines to which it is connected) as a graph in which all branches can show such a defect; see Figure 2.

Opens at locations OCx and OCxc will show a *complementary* fault behavior due to the symmetric structure of the memory cell. For that reason, one needs to

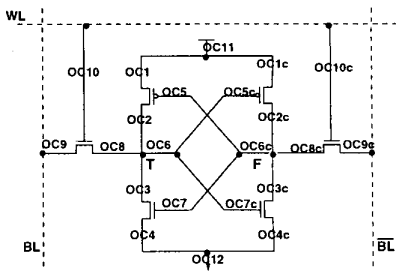

Figure 2. Opens within a cell

simulate opens at OCx locations only. From these, the behavior of the opens OCxc can be derived. The first block of Table 1 lists the OCx opens.

Table 1. List of opens

Name	Description
OC1/OC2	Source/drain of pull-up at true side broken
OC3/OC4	Drain/source of pull-down at true side broken
OC5	Gate of pull-up at true side broken
OC6	Cross coupling at true side broken
OC7	Gate of pull-down at true side broken
OC8	Pass transistor connection to T broken
OC9	Pass transistor connection to bit line broken
OC10	Gate of pass transistor at true side broken
OC11/OC12	V_{cc}/V_{ss} path of the cell broken
OB_w	The bit line BL at the write side broken
OB_r	The bit line BL at the read side broken
OW	The word line WL broken

2.1.2 Opens at bit lines and word lines

Bit lines and word lines are connected to many cells. Therefore, an open at a bit line or at a word line can influence the behavior of the operations applied to all these cells. In the following, first opens at bit lines will be discussed and thereafter at word lines.

Opens at bit lines: If we consider that the memory cell array is located between the read and the write circuit, then the opens at bit lines can occur at the following locations: (a) An open between the cell and the write circuit (denoted as OB_w), and (b) An open between the cell and the read circuit (denoted as OB_r). Since there is a pair of bit lines connected to each cell, 4 opens at bit lines can exist; two at the side of the write circuit and two at the side of read circuit. However, one can be limited to simulate only two opens (e.g., OB_w and OB_r at BL), while the behavior of the other opens (i.e., opens at $\overline{BL}$) can be derived. This is because the opens at $\overline{BL}$ have a complementary behavior to opens at BL.

Opens at word lines: Since the opens at the pass transistor gates have already been covered as opens within a cell, the only remaining opens are those in the

common parts of the word lines. We will define OW as an open at the word line WL.

The bottom block of Table 1 lists the minimal set of opens at bit lines and word lines that has to be simulated.

2.2 Definition and location of shorts

The shorts can be classified as shorts within a cell (denoted as SC) and shorts at bit lines (denoted as SB) and at word lines (denoted as SW). Power shorts (i.e., shorts between V_{cc} and V_{ss}) are excluded, since they do not belong to the class of memory cell array faults; they impact the behavior of the whole circuit.

2.2.1 Shorts within a cell

In this case, the cell is considered without bit lines and word lines. Due to the symmetric structure of the memory cell, a short at F will show a complementary behavior to short at T; see Figure 1. Each short is defined as a pair of nodes in which one node is V_{cc} or V_{ss}. Table 2 shows the possible shorts within a cell (SCs); shorts with a complementary fault behavior are grouped together in the same row.

Table 2. List of possible shorts

Shorts		Comp. beh.
SC1	T-V_{cc}	F-V_{cc}
SC2	T-V_{ss}	F-V_{ss}
SB1	BL-V_{cc}	$\overline{BL}$-V_{cc}
SB2	BL-V_{ss}	$\overline{BL}$-V_{ss}
SW1	WL-V_{cc}	
SW2	WL-V_{ss}	

2.2.2 Shorts at bit lines and word lines

The cells belonging to the same column (or the same row) are connected to the same bit lines (or the same word lines). Therefore, shorts at bit lines and word lines can affect the behavior of all operations performed to these cells. Table 2 lists the possible shorts at bit lines (SBs) and at word lines (SWs).

2.3 Definition and location of bridges

A bridge in the SRAM memory cell array can connect any arbitrary pair of nodes. This section enumerates the total set of bridges of interest. The assumption is made that the nodes have to be located close to each other, such that bridges can be divided into two groups:

Bridges within a cell: All bridges connecting two nodes of the same cell, including the pair of bit lines and the word line to which it is connected.

Bridges between cells: All bridges connecting nodes of adjacent cells, including the bit lines and the word lines to which the cells are connected.

2.3.1 Bridges within a cell

To define all possible bridges within a cell, the cell has to be considered as a graph in which each node can be connected to another by a bridge. Since each cell consists of five nodes n_i ($n_i \in \{T, F, BL, \overline{BL}, WL\}$), there are $C_2^5 = \frac{5!}{2!*3!} = 10$ possible bridges; see Table 3. Note that bridges with a complementary behavior (Comp. beh.) are grouped together in the same row.

Table 3. List of bridges within a cell

Bridge		Comp. beh.
BC1	T-F	
BC2	T-BL	F-$\overline{BL}$
BC3	T-$\overline{BL}$	F-BL
BC4	T-WL	F-WL
BC5	BL-$\overline{BL}$	
BC6	BL-WL	$\overline{BL}$-WL

2.3.2 Bridges between cells

To establish all possible bridges between adjacent cells, the configuration shown in Figure 3 will be considered. It consists of of four cells: C_1, C_2, C_3 and C_4. Note that the adjacent cells can belong to the same column, the same row, or to the same diagonal. The cells C_1 and C_3, as well as the cells C_2 and C_4, are adjacent in the same row and therefore have common word lines (i.e., $WL1$, respectively $WL2$); while the cells C_1 and C_2 (as well as the cells C_3 and C_4) are adjacent in the same column and therefore have common bit lines.

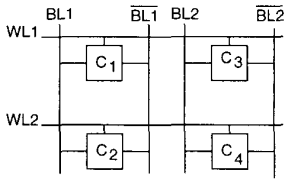

Figure 3. Four cell configuration

1. Bridges between cells in same row

To define all possible bridges between adjacent cells in the same row (denoted as rBCCs), only C_1 and C_3 have to be considered; see Figure 3. Both $C1$ and C_3 consist of five nodes: C_1 consists of $n_1 \in \{T1, F1, BL1, \overline{BL1}, WL1\}$; while C_3 consists of $n_2 \in \{T3, F3, BL2, \overline{BL2}, WL1\}$. Since both cells have a common word line, only the true/false node (T1, F1) or bit lines of C_1 can form a bridge with the true/false node or with the bit lines of C_3. Therefore, there are 16 possible bridges, $n_1 - n_2$, between the two cells; whereby n_1 can be one node of the set $\{T1, F1, BL1, \overline{BL1}\}$, and n_2 of $\{T3, F3, BL2, \overline{BL2}\}$;

see rBCCs bridges in Table 4. Bridges with a complementary (Comp. beh.), an interchanged (Inter. beh.), or an interchanged complementary behavior (I.C. beh.) are grouped together in the same row.

Table 4. Bridges between adjacent cells

Bridge		Comp. beh.	Inter. beh.	I.C. beh.
rBCC1	T1-T3	F1-F3		
rBCC2	T1-F3	F1-T3		
rBCC3	T1-$\overline{BL2}$	F1-$\overline{BL2}$	$\overline{BL1}$-T3	$\overline{BL1}$-F3
rBCC4	T1-$\overline{BL2}$	F1-$\overline{BL2}$	$\overline{BL1}$-T3	$\overline{BL1}$-F3
rBCC5	$\overline{BL1 - BL2}$	$\overline{BL1 - BL2}$		
rBCC6	$\overline{BL1 - BL2}$		$\overline{BL1 - BL2}$	
cBCC1	T1-T2	F1-F2		
cBCC2	T1-F2	F1-T2		
cBCC3	T1-$\overline{WL2}$	F1-$\overline{WL2}$	$\overline{WL1}$-T2	$\overline{WL1}$-F2
cBCC4	$\overline{WL1}$-$\overline{WL2}$			
dBCC1	T1-T4	F1-F4		
dBCC2	T1-F4	F1-T4		

2. Bridges between cells in same column

To define all possible bridges between adjacent cells in the same column, (denoted as cBCCs), only C_1 and C_2 have to be considered; see Figure 3. Both $C1$ and C_2 consist of five nodes: C_1 consists of $n_1 \in \{T1, F1, BL1, \overline{BL1}, WL1\}$, while C_2 consists of $n_2 \in \{T2, F2, BL1, \overline{BL1}, WL2\}$. Note that the two cells share the same bit lines. Therefore, there are only 9 possible bridges, $n_1 - n_2$, between C_1 and C_2; whereby $n_1 \in \{T1, F1, WL1\}$ and $n_2 \in \{T2, F2, WL2\}$; see cBCCs bridges in Table 4. A bridge between the bit lines and the nodes T2 or F2 is excluded since it belongs to bridges within a cell, which are already considered in Section 2.3.1.

3. Bridges between diagonal cells

The possible bridges, denoted as dBCCs, between cells belonging to the same diagonal (i.e., C_1 and C_4 of Figure 3) consist only of four bridges; see dBCCs bridges in Table 4. All other bridges between the nodes of C_1 and the nodes of C_4 are already considered in 1 and 2 above; this is because C_4 has the same word line as C_2 and the same bit lines as C_3.

3 Experimental results

Before listing the simulation results, the notation describing the faults will be introduced.

3.1 Notation of faults

The faults in single port SRAMs can be divided into faults involving a *single cell*, and faults involving *two cells*. In order to describe these faults, a compact notation (referred as *fault primitive (FP)*) will be used.

- $< S/F/R >$ (or $< S/F/R >_v$): denotes a FP involving a *single cell*; the cell c_v (victim cell) used to sensitize a fault is the same as where the fault appears. S describes the *sensitizing* operation or state; $S \in \{0, 1, w0, w1, w\uparrow, w\downarrow, r0, r1, \forall\}$, whereby 0 denotes a *zero* value, 1 denotes a *one* value, $w0$ ($w1$) denotes a write 0 (1) operation, $w\uparrow$ ($w\downarrow$) denotes an up (down) transition write operation, $r0$ ($r1$) denotes a read 0 (1) operation, and $\forall$ denotes any operation ($\forall \in \{0, 1, w1, w0, w\uparrow, w\downarrow, r1, r0\}$); if the fault effect of S appears after a time T, then the sensitizing operation is given as S_T.
- $< S_a; S_v/F/R >$ (or $< S_a; S_v/F/R >_{a,v}$): denotes a FP involving *two cells*; S_a describes the sensitizing operation or state of the *aggressor cell (a-cell)*; while S_v describes the sensitizing operation or state of the *victim cell (v-cell)*. The a-cell (c_a) is the cell sensitizing a fault in an other cell called the v-cell (c_v). The set of S_i is defined as: $S_i \in \{0, 1, X, w0, w1, w\uparrow, w\downarrow, r0, r1\}$ ($i \in \{a, v\}$), whereby X is the don't care value $X \in \{0, 1\}$.

In both notations, F describes the value of the *faulty* cell (v-cell); $F \in \{0, 1, \uparrow, \downarrow, ?\}$, whereby $\uparrow$ ($\downarrow$) denotes an up (down) transition, and ? denotes an undefined logical value. R describes the logical value which appears at the output of the SRAM if the sensitizing operation applied to the v-cell is a *read* operation; $R \in \{0, 1, ?, -\}$, whereby ? denotes an undefined or random logical value. An undefined logical value can occur if the voltage difference between the bit lines (used by the sense amplifier) is very small. A '-' in R means that the output data is not applicable in that case; e.g., if $S = w0$, then no data will appear at the memory output, and therefore R is replaced by a '-'.

3.2 Simulation results

In the representation of the simulation results, the following terminology will be used:

- *Strong fault:* This is a memory fault that can be **fully** *sensitized* by any operation; i.e., a write or a read operation fails. That means that the state of the cell is incorrectly changed, can not be changed, or that the sense amplifier returns an incorrect result.
- *Weak fault:* This is a fault which is **partially** sensitized by an operation; e.g., a defect that creates a small disturbance of the voltage of the true node of the cell. Note that in the presence of a weak fault, all operations (read and write) pass correctly.

The simulation has been done for all opens, shorts, and bridges by examining the resistance range from 0 to ∞, and by using a simulation methodology similar to that described in [5]. It examines all possible operations in the presence of an open, a short or a bridge.

Each faulty behavior is reported in terms of a fault primitive (FP); if a strong fault is sensitized, then the FP notation introduced Section 3.1 is used to describe it. If a fault is only partially sensitized (e.g., weak fault) then the fault is denoted as wF.

In order to save space, only the simulation results for bridges between cells will be presented in this paper. The simulation results for opens, shorts and bridges within a cell are given in [9]. Table 5 lists the simulation results for bridges between cells in the same row (rBCCs), between cells in the same column (cBCCs), and between cells on the same diagonal (dBCCs). The first column in the table gives the name of the bridge; the second, lists the resistance regions[1] ordered in increasing resistance values; the third and the fourth columns give the FP sensitized by the simulated spot defect, and the derived complementary FP (if applicable), respectively. The fifth column lists the class of the sensitized FP; i.e., an FP involving a single-cell (FP1) and an FP involving two cells (FP2). The table shows clearly that the sensitized FP is *strongly dependent* on the resistance value of the defect. Note that the FPs caused by the bridge cBCC1 and the bridge dBCC1 are the same, as well as the FPs caused by the bridges cBCC2 and dBCC2.

4 Functional faults models

The faults caused by spot defects expressed in terms of FPs are translated into *functional fault models (FFMs)*; see column 6 in Table 5. *A FFM is defined as a non empty set of FPs.* The FFMs are divided into FFMs involving a single cell (FFM1s) and FFMs involving two cells (FFM2s). The FFM1s consist of single-cell FPs and have the property that the cell used for sensitizing the fault is the same cell as where the fault appears; while the FFM2s consist of two-cell FPs and have the property that the application of an operation to a cell c_a (or the state of the cell c_a) has as a consequence that a fault will be sensitized in another cell c_v.

4.1 The FFM1 fault class

In order to describe this fault class, the notation introduced in Section 3.1 will be used. Based on the fault behaviors found by simulating opens, shorts and bridges, the following FFM1s have been derived.

Stuck-at Fault (SAF): the logic value of a cell is always '0' or always '1'. The SAF consists of two FPs: $< \forall/0/- >$, and $< \forall/1/- >$; it can be caused by defects like: (a) Cross coupling between one of the nodes of the cell and the opposite invertor is broken (OC6), (b) Short at one of the nodes of the cell (SC1, SC2).

[1]The exact values are design specific and Intel proprietary

Transition Fault (TF): the cell fails to undergo a transition ($0 \rightarrow 1$ or $1 \rightarrow 0$) when it is written. The TF consists of two FPs: $< w \uparrow /0/- >$ and $< w \downarrow /1/- >$; it can be caused by defects like: (a) Pass transistor connection of the cell broken (OC8, OC9), (b) Short at one of the bit lines of the cell (SB1, SB2).

Stuck-Open Fault (SOF): SOF is defined as an inaccessible cell; i.e., the cell fails to undergo both transitions when it is written, and every read operation performed to the cell returns random or undefined data. The *SOF* consists of the following FPs which occur simultaneously: $< w \uparrow /0/- >$, $< w \downarrow /1/- >$, and $< rx/x/? >$; it can be caused by: (a) A broken word line (OW) and (b) A word line shorted to V_{ss} (SW2).

Data Retention Fault (DRF): the cell fails to retain its logic value after some period of time T. The DRF consists of four FPs: $< 1_T/ \downarrow /- >$, $< 0_T/ \uparrow /- >$, $< 1_T/?/- >$, and $< 0_T/?/- >$; it can be caused by: (a) Source/drain/gate of the pull-up transistor of the cell broken (OC1, OC2, OC5), (b) V_{cc}/V_{ss} path broken (OC11, OC12).

Read Destructive Fault (RDF): a rx operation performed to the cell changes the data in the cell into $\overline{x}$ and returns the logic value $\overline{x}$. The RDF consists of two FPs: $< r0/ \uparrow /1 >$ and $< r1/ \downarrow /0 >$; it can be caused by defects like: (a) Drain/source of the pull-down transistor of the cell broken (OC3, OC4), (b) Bit line shorted to V_{ss} (SB2), (c) A bridge between a bit line and a word line (BC6).

Deceptive Read Destructive Fault (DRDF): a rx operation performed to the cell changes the data in the cell into $\overline{x}$ while it returns the logic value x. The DRDF consists of two FPs: $< r0/ \uparrow /0 >$ and $< r1/ \downarrow /1 >$; it can be caused by the following defects: (a) Drain/source of the pull-down transistor of the cell broken (OC3,OC4), (b) Node of the cell shorted to V_{ss} (SC2).

Incorrect Read Fault (IRF): a rx operation performed to a cell returns the logic value $\overline{x}$, while the state of the cell is not changed. The IRF consists of two FPs: $< r0/0/1 >$ and $< r1/1/0 >$; it can be caused by (a) A bit line shorted to V_{ss} (SB2), (b) A bridge between the node of the cell and a bit line (BC3), (c) A bridge between a bit line and a word line (BC6).

Random Read Fault (RRF): a rx operation performed to a cell returns a random value, while the state of the cell is not changed. This is due to the fact that the voltage difference between the bit lines is too small. The RRF consists of two FPs: $< r0/0/? >$ and $< r1/1/? >$; it can be caused by defects like: (a) A connection to the pass transistor broken (OC8, OC9), (b) Bit line at the read circuit side broken (OB_r).

Table 5. Simulation results for the bridges between cells (BCCs)

Name	R_{br} region	Fault behavior	Comp. behavior	Class	FFM
rBCC1	Region *I*	$< 0; X/0/- >_{i,j}$	$< 1; X/1/- >_{i,j}$	FP2	CF_{st}
	Region *II*	$< 0; r1/\downarrow/0 >_{a,v}$	$< 1; r0/\uparrow/1 >_{a,v}$	FP2	CF_{rd}
		$< r0; 1/\downarrow/- >_{a,v}$	$< r1; 0/\uparrow/- >_{a,v}$	FP2	CF_{ds}
	Region *III*	wF	wF	-	-
rBCC2	Region *I*	$< 0; X/1/- >_{i,j}$	$< 1; X/0/- >_{i,j}$	FP2	CF_{st}
		$< 1; X/0/- >_{j,i}$	$< 0; X/1/- >_{j,i}$	FP2	CF_{st}
	Region *II*	$< 1; r1/\downarrow/0 >_{a,v}$	$< 0; r0/\uparrow/1 >_{a,v}$	FP2	CF_{rd}
		$< r0; 0/\uparrow/- >_{a,v}$	$< r1; 1/\downarrow/- >_{a,v}$	FP2	CF_{ds}
	Region *III*	wF	wF	-	-
rBCC3	Region *I*	$< \forall/1/- >_v$	$< \forall/0/- >_v$	FP1	SAF
	Region *II*	$< w\downarrow/1/- >_v$	$< w\uparrow/0/- >_v$	FP1	TF
		$< w\uparrow/0/- >_v$	$< w\downarrow/1/- >_v$	FP1	TF
	Region *III*	$< w0; 1/\downarrow/- >_{a,v}$	$< w1; 0/\uparrow/- >_{a,v}$	FP2	CF_{ds}
		$< 0; r0/0/1 >_{a,v}$	$< 1; r0/0/1 >_{a,v}$	FP2	CF_{ir}
	Region *IV*	wF	wF	-	-
rBCC4	Region *I*	$< \forall/1/- >_v$	$< \forall/0/- >_v$	FP1	SAF
	Region *II*	$< w\downarrow/1/- >_v$	$< w\uparrow/0/- >_v$	FP1	TF
		$< w\uparrow/0/- >_v$	$< w\downarrow/1/- >_v$	FP1	TF
	Region *III*	$< w1; 1/\downarrow/- >_{a,v}$	$< w0; 0/\uparrow/- >_{a,v}$	FP2	CF_{ds}
		$< 0; r0/0/1 >_{a,v}$	$< 1; r1/1/0 >_{a,v}$	FP2	CF_{ir}
	Region *IV*	wF	wF	-	-
rBCC5	Region *I*	$< w0; 1/\downarrow/ >_{a,v}$	$< w1; 0/\uparrow/-/ >_{a,v}$	FP2	CF_{ds}
	Region *III*	wF	wF	-	-
rBCC6	Region *I*	$< w0; 0/\uparrow/ >_{a,v}$		FP2	CF_{ds}
		$< w1; 1/\downarrow/ >_{a,v}$		FP2	CF_{ds}
	Region *II*	wF	wF	-	-
cBCC1	Region *I*	$< 0; X/0/- >_{i,j}$	$< 1; X/1/- >_{i,j}$	FP2	CF_{st}
	Region *II*	$< 0; r1/\downarrow/- >_{a,v}$	$< 1; r0/\uparrow/- >_{a,v}$	FP2	CF_{rd}
	Region *III*	wF	wF	-	-
cBCC2	Region *I*	$< 0; X/1/- >_{i,j}$	$< 1; X/0/- >_{i,j}$	FP2	CF_{st}
	Region *II*	$< 1; r1/\downarrow/0 >_{a,v}$	$< 0; r0/\uparrow/1 >_{a,v}$	FP2	CF_{rd}
	Region *III*	wF	wF	-	-
cBCC3	Region *I*	$< \forall/0/- >_v$	$< \forall/1/- >_v$	FP1	SAF
	Region *II*	$< r1/\downarrow/0 >_v$	$< r0/\uparrow/1 >_v$	FP1	RDF
	Region *III*	$< r1/\downarrow/1 >_v$	$< r0/\uparrow/0 >_v$	FP1	DRDF
	Region *IV*	wF	wF	-	-
cBCC4	Region *I*	$< w\uparrow/0/- >_v$		FP1	TF
		$< w\downarrow/1/- >$		FP1	TF
dBCC1	Region *I*	$< 0; X/0/- >_{i,j}$	$< 1; X/1/- >_{i,j}$	FP2	CF_{st}
	Region *II*	$< 0; r1/\downarrow/- >_{a,v}$	$< 1; r0/\uparrow/- >_{a,v}$	FP2	CF_{rd}
	Region *III*	wF	wF	-	-
dBCC2	Region *I*	$< 0; X/1/- >_{i,j}$	$< 1; X/0/- >_{i,j}$	FP2	CF_{st}
	Region *II*	$< 1; r1/\downarrow/0 >_{a,v}$	$< 0; r0/\uparrow/1 >_{a,v}$	FP2	CF_{rd}
	Region *III*	wF	wF	-	-

Undefined State Fault (USF): a wx or a rx operation performed to a cell brings the cell into an undefined state; the read operation then returns a random value. The USF consists of four FPs: $< w0/?/? >$, $< w1/?/? >$, $< r0/?/? >$ and $< r1/?/? >$; it can be caused by an open at the V_{ss} line of the cell (OC13), or by a bridge between the nodes of the same cell (BC1).

4.1.1 The FFM2 fault class

Based on the fault behaviors found by simulating the SDs, the following FFM2s have been derived.

Disturb Coupling Fault (CF_{ds}): a v-cell undergoes a transition due to a wx or rx operation applied to the a-cell. The CF_{ds} consists of eight FPs: $< w0; 0/\uparrow/- >$, $< w0; 1/\downarrow/- >$, $< w1; 0/\uparrow/- >$, $< w1; 1/\downarrow/- >$, $< r0; 0/\uparrow/- >$, $< r0; 1/\downarrow/- >$, $< r1; 0/\uparrow/- >$, and $< r1; 1/\downarrow/- >$; it can be caused by defects like: (a) Pass transistor gate floating (OC10), (b) Word line shorted to V_{cc} (SW1), (c) A bridge between adjacent cells in the same row (rBCC1 through rBCC6).

State Coupling Fault (CF_{st}): the v-cell is forced to a certain logic value (0 or 1), only if the a-cell is in a given state (0 or 1). The CF_{st} consists of four FPs: $< 1; X/0/->$, $< 1; X/1/->$, $< 0; X/0/->$, and $< 0; X/1/->$; it can be caused by a bridge between nodes of adjacent cells (rBCC1, rBCC2, cBCC1, cBCC2, dBCC1, dBCC2).

Incorrect Read Coupling Fault (CF_{ir}): a read operation applied to the v-cell returns an incorrect value if the a-cell is in a certain state. Note that in this case, the state of the v-cell is not changed. The CF_{ir} consists of four FPs: $< 0; r0/0/1 >$, $< 0; r1/1/0 >$, $< 1; r0/0/1 >$, and $< 1; r1/1/0 >$; it can be caused by: (a) A bridge between a node of a cell and its bit line (BC2, BC3), (b) A bridge between a node of a cell and a bit line of an adjacent cell (rBCC3, rBCC4).

Random Read Coupling Fault (CF_{rr}): a read operation applied to the v-cell returns a random value if the a-cell is in a certain state. The CF_{rr} consists of four FPs: $< 0; r0/0/? >$, $< 0; r1/1/? >$, $< 1; r0/0/? >$ and $< 1; r1/1/? >$; it can be caused by: (a) Pass transistor gate floating (OC10), (b) Word line broken (OW), (c) A bridge between the node of the cell and its bit line (BC2, BC3).

Read Destructive Coupling Fault (CF_{rd}): a read operation applied to the v-cell causes a transition in the v-cell and returns an incorrect value, if the a-cell is in a given state. The CF_{rd} consists of four FPs: $< 0; r0/\uparrow/1 >$, $< 0; r1/\downarrow/0 >$, $< 1; r0/\uparrow/1 >$, and $< 1; r1/\downarrow/0 >$; it can be caused by a bridge between nodes of adjacent cells (rBCC1, rBCC2, cBCC1, cBCC2, dBCC1, dBCC2).

5 March SR for realistic FFMs

This section first describes the new march tests **March SR** and **March SRD**, after which a comparison is made with well-known other march tests.

5.1 March SR and March SRD

A new test referred to as *March SR*, test for *simple* (i.e., not linked) *realistic* faults, will be introduced; linked faults are faults which change the behavior of other faults with the possibility of masking. March SR is of interest when some extra test time can be tolerated for covering *all* realistic faults discussed in Section 4.

In order to describe March SR, march notation will be used. A complete march test is delimited by the '{...}' bracket pair; while a march element is delimited by the '(...)' bracket pair. The march elements are separated by semicolons, and the operations within a march element are separated by commas. Note that all operations of a march element are performed at a certain address, before proceeding to the next address. The latter can be done in either one of two address orders: an increasing ($\Uparrow$) or a decreasing ($\Downarrow$) address order. When the address order is not relevant, the symbol $\Updownarrow$ will be used.

The test is shown in Figure 4; it has a test length of $14n$. It detects all FFMs with a deterministic data output at the sense amplifier, except DRFs. In addition, it may probabilistically detect the FFMs with a random data outputs since each cell is read with different data values within a single march element (see march elements M_1 and M_4). Note that the symmetrical structure of the test make it better suitable for BIST implementation [10].

- All SAFs, IRFs, and RDFs are detected since from each cell a 0 and a 1 is read.
- All TFs are detected because each cell is read after an $w\uparrow$ and a $w\downarrow$ operation; see M_1 and also M_4.
- All DRDFs are detected because two *successive* read operations are applied to each cell (see M_2 and M_5); the first operation will sensitize the fault and the second will detect it.
- All SOFs are detected. The detection of SOFs requires that a test has to consist of a march element which reads the values x and $\overline{x}$ from a cell [13]. This condition is satisfied by M_1 (and M_4).
- All CF_{st}s, CF_{ir}s, CF_{rd}s, and CF_{ds}s are detected. March elements M_1 and M_4 contain all sensitizing operations; all states are entered (for CF_{st}s), and a read operation is performed after entering each state (to detect CF_{st}s, CF_{ir}s and CF_{rd}s), and they contain all sensitizing operations (read and write) for CF_{ds}s. When the value of the fault effect is 1, then the CF_{ds}s will be detected by the $r0$ of M_1 if the v-cell has a higher address than the a-cell, and by the $r0$ of M_2 if the v-cell has a lower address than the a-cell. M_4 and M_5 detect the CF_{ds}s when the value of the fault effect is 0.

$$\{\ \underset{M_0}{\Downarrow(w0)}\ ;\ \underset{M_1}{\Uparrow(r0,w1,r1,w0)}\ ;\ \underset{M_2}{\Downarrow(r0,r0)}\ ;\ \underset{M_3}{\Uparrow(w1)}\ ;\ \underset{M_4}{\Downarrow(r1,w0,r0,w1)}\ ;\ \underset{M_5}{\Uparrow(r1,r1)}\ \}$$

Figure 4. March SR

The detection of DRFs requires bringing the cell in a certain state, adding a certain delay, and thereafter reading the cell (this has to be done for both logic states of the cell). March SR can be extended in order to additionally detect DRFs. The result is shown in Figure 5, and referred as *March SRD*. The two inserted delay elements permit the detection of the two FPs of DRF.

$$\{\ \underset{M_0}{\Downarrow(w0)}\ ;\ \underset{M_1}{\Uparrow(r0,w1,r1,w0)}\ ;Del;\ \underset{M_2}{\Downarrow(r0,r0)}\ ;\ \underset{M_3}{\Uparrow(w1)}\ ;\ \underset{M_4}{\Downarrow(r1,w0,r0,w1)}\ ;Del;\ \underset{M_5}{\Uparrow(r1,r1)}\ \}$$

Figure 5. March SRD

5.2 Comparison with other tests

Many memory test algorithms were developed to cover FFMs most of which had a theoretical origin. The traditional ad-hoc tests have been used to screen the faulty devices. Walking 1/0, GALPAT, Butterfly, Zero-one test, and Checkerboard are widely known tests. However, the time complexity of the first two tests is completely unacceptable for any serious testing purpose; while the fault coverage of the last three tests is not acceptable industrially.

March tests have been introduced to detect TFs, *Inversion Coupling Faults (CF_{in}s), Idempotent Coupling Faults CF_{id}s*, as well as SAFs. A CF_{in} is defined as: an up (or down) transition write operation in the a-cell causes an inversion in the v-cell; i.e., the v-cell flips to 0 if its content was 1, and flips to 1 if its content was 0. A CF_{id} is defined as: an up (or down) transition write operation in the a-cell forces a certain fixed value, 0 or 1, in the v-cell. Fortunately, most of the introduced march tests also detect most of the new FFMs of Section 4. Table 6 summarizes the fault coverage of several march tests. In the table, the sign '+' denotes that the test detects the correspondenting FFM; the sign '-' denotes that the FFM is not detected; the sign '-+' denotes that the FFM is detected

probabilistically, however with a low probability; the sign '-++' denotes the same as the sign '-+', but with a high probability of detection. Note that the FFMs CF_{in} and CF_{id} are not included in the table since they are considered *not* to be realistic; this is because they have not been found using the circuit simulation. In [3], it has also been shown that the empirical evidence does not support the occurrence of CF_{in}s.

Table 6. Fault coverage for different tests

FFM	March Tests				
	MATS+ (5n)	March C- (10n)	March B (17n)	IFA 13n (13n)	March SRD (14n)
SAF	+	+	+	+	+
TF	-	+	+	+	+
SOF	-	-	+	+	+
RDF	+	+	+	+	+
DRDF	-	-	-	-	+
IRF	+	+	+	+	+
DRF	-	-	-	+	+
RRF	-+	-+	-++	-++	-++
USF	-+	-+	-++	-++	-++
CF_{ds}	-	+	-	+	+
CF_{st}	-	+	-	+	+
CF_{ir}	-	+	-	+	+
CF_{rd}	-	+	-	+	+
CF_{rr}	-	-+	-	-++	-++

The Modified Algorithm Test Sequence (MATS+) [11] detects all SAFs, and has a test length of $5n$, whereby n is the memory size. March C- [12] was introduced to detect SAFs, TFs, as well as CF_{in}s and CF_{id}s. The test, which has a test length of $10n$, also detects the CF_{st}s [3] and RDFs [5]. In addition, it can be shown that the test also detects the following new FFMs of Section 4: IRFs, CF_{ds}s (which assume disturbs by write as well as by *read* operations), CF_{rd}s, and CF_{ir}s; and may detect RRFs, USFs, and CF_{rr}s. However, March C- can not detect DRDFs, SOFs, and DRFs.

March B [13], which is an extension of March A [13], was designed to detect *linked* CF_{in}s and CF_{id}s. The test "IFA 13n" [3] was the first test designed to target SOFs and DRFs; it has two delay elements and four extra read operations; two are redundant (i.e., they can be removed without impacting the fault coverage).

6 Conclusions

In this paper a complete analysis of all spot defects in an industrial single port SRAM design has been performed, based on circuit simulation. The transformation of the electrical faults, caused by the defects, into functional fault models (FFMs) has been presented. It has been shown that the FFM is strongly dependent on the resistance value of the defect. The existence of the traditional memory fault models has been verified; no *inversion coupling faults* and no *idempotent coupling faults* have been found. Instead, three new FFMs involving a single cell (FFM1s), and three involving two cells (FFM2s) have been introduced; the three FFM1s consist of the Incorrect Read Fault (IRF), the Random Read Fault (RRF), and the Undefined State Fault (USF); while the three FFM2s consist of the Incorrect Read Coupling Fault (CF_{ir}), the Read Destructive Coupling Fault (CF_{rd}), and the Random Read Coupling Fault (CF_{rr}).

Finally a new march test detecting all FFMs (re)introduced in this paper, with a test length of $14n$, has been given; its fault coverage has been compared with other well-known tests.

Acknowledgment

We express our gratitude to David Eastwick, Mike Rodgers, Mike Spica, and Greg Tollefson from Intel corporation for providing extensive information regarding the tools as well as the SRAM design.

References

[1] M. Inoue, *et. al*, "A New Test Evaluation Chip for Lower Cost Memory Tests", *IEEE Design and Test of Computers*, **10**(1), pp. 15-19, March 1993.

[2] A.J. van de Goor and J. de Neef, "Industrial Evaluation of DRAMs Tests", *In Proc. of Design Automation and Test in Europe*, pp. 623-630, March 1999.

[3] R. Dekker, *et al.*, "A Realistic Fault Model and Test Algorithms for Static Random Access Memories", *IEEE Trans. on Computers*, **C9**(6), pp. 567-572, 1990.

[4] S. Naik, *et. al*, "Failure Analysis of High Density CMOS SRAMs", *IEEE Design and Test of Computers*, **10**(1), pp. 13-23, June 1993.

[5] R.D. Adams, "Extensions of Static Random Access Memory Fault Modeling and Examination of Patterns for Fault Detections", *M.S. Thesis, Thayer School of Engineering*, Dartmouth, 1996.

[6] V. Kim and T. Chen, "Assessing SRAM Test Coverage for Sub-Micron CMOS Technologies", *In Proc. of IEEE VLSI Test Symposium*, pp.24-30, Monterey, CA., USA, April 1997.

[7] T.M. Mak, *et. al*, "Cache RAM Inductive Fault Analysis with Fab Defect Modeling", *In Proc. of IEEE International Test Conference*, pp. 862-871, Oct. 1998.

[8] A.J. van de Goor and J.E. Simonse, "Defining Appropriate SRAM Resistive Defects and their Simulation Stimuli", *In Proc. of Asian Test Symposium*, pp. 33-40, Nov. 1999.

[9] S. Hamdioui and A.J. van de Goor, "An Industrial Analysis of Spot Defects in SRAMs: Realistic Fault Models and Tests", *Submitted to IEEE Trans. on Computers.*

[10] D.R. Aadsen, *et al.*, "Automated BIST for Regular Structures Embedded in ASIC Devices", *AT&T Technical Journal*, **69**(3), pp. 97-105, 1990.

[11] R. Nair, "An Optimal Algorithm for Testing Stuck-At-Faults in Random Access Memories", *IEEE Trans. on Computers*, **C28**(3), pp. 258-261, 1979.

[12] A.J. van de Goor, "Testing Semiconductor Memories, Theory and Practice", *ComTex Publishing, Gouda, The Netherlands*, 1998. Web: http://cardit.et.tudelft.nl/~ vdgoor

[13] D.S. Suk and S.M. Reddy, "A March Test for Functional Faults in Semiconductor Random-Access Memories", *IEEE Trans. on Computers*, **C30**(12), pp. 982-985, 1981.