Testing Static and Dynamic Faults in Random Access Memories

Said Hamdioui[1,2] Zaid Al-Ars[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 Computer Engineering, Mekelweg 4, 2628 CD Delft, The Netherlands
E-mail: said@ce.et.tudelft.nl

Abstract

The ever increasing trend to reduce DPM levels of memories requires tests with very high fault coverages. The very important class of dynamic fault, therefore cannot be ignored any more. It will be shown in this paper that conventional memory tests constructed to detect the static faulty behavior of a specific defect do not necessarily detect its dynamic faulty behavior; which has been shown to exist. The dynamic fault behavior can take place in the absence of the static fault behavior. The paper therefore also presents new memory tests derived to target the dynamic fault class.

Key words: *static faults, dynamic faults, fault primitives, memory tests, fault coverage.*

1 Introduction

The cost of testing memories increases rapidly with every new generation of memory chips [8]. 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.

Researchers studying the faulty behavior of memory devices have been defining *functional fault models (FFMs)* and developing tests to target them [1, 4, 5, 6, 10, 11, 12, 13, 15, 19, 21]. Most of the work published on fault modeling focuses on faults sensitized by performing *at most one operation*; e.g., a write operation sensitizes a fault. These FFMs are called *static faults*. Experimental analysis of DRAMs, based on defect injection and SPICE simulation, shows that another type of faulty behavior can take place in the absence of static faults [2, 3]. This faulty behavior requires more than one operation to be sensitized. For example, a write operation followed immediately by a read operation causes the cell to flip; however, if only a write or a read operation is performed, the cell will not flip. Faults requiring *more than one operation sequentially* in order to be sensitized are called *dynamic faults*. Most used tests are currently designed for static faults, and therefore may not be able to detect dynamic faults. All that indicates the importance of dynamic faults.

This paper uses an analytical approach for analyzing dynamic faults; it is organized as follows. Section 2 introduces the concept of *fault primitives* that will be used to define dynamic faults in Section 3. Section 4 describes the importance of dynamic faults based on some simulation results. Section 5 shows that tests designed for static faults do not necessary detect dynamic faults; Section 6 introduces tests for dynamic faults, while Section 7 ends with the conclusions.

2 Fault primitives

By performing a number of memory operations and observing the behavior of any component functionally modeled in the memory, functional faults can be defined as the deviation of the observed behavior from the specified one under the performed operation(s). Therefore, the two basic ingredients to any fault model are: (a) A list of performed memory operations, and (b) A list of corresponding deviations in the observed behavior from the expected one. Any list of performed operations on the memory is called an *operation sequence*. An operation sequence that results in a difference between the observed and the expected memory behavior is called a *sensitizing operation sequence (S)*. The observed memory behavior that deviates from the expected one is called the *faulty behavior (F)*.

In order to specify a certain fault, one has to specify the S, together with the corresponding faulty behavior F. The combination of S and F for a given memory failure is called a *Fault Primitive (FP)* [20], and 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 (e.g., the cell flips from 0 to 1), while R describes the logic output level of a read operation (e.g., 0) in case S

is a read operation. E.g., the up transition fault is described as $< 0w1/0/- >$.

The concept of FPs allows for establishing a complete framework of memory faults, since for all allowed operation sequences in the memory, one can derive all possible types of faulty behavior. In addition, the concept of a FP makes it possible to give a precise definition of a *functional fault model (FFM)* [20]: *a functional fault model is a non-empty set of fault primitives*.

Let $\#O$ be defined as the number of different operations performed *sequentially* in S. For example, if a single read operation applied to a certain cell causes that cell to flip, then $\#O = 1$. Depending on $\#O$, FPs can be divided into *static* and *dynamic* faults:

- *Static faults*: These are FPs sensitized by performing *at most* one operation: $\#O \leq 1$. For example, the state of the cell is always stuck at *one* ($\#O = 0$), a read operation to a certain cell causes that cell to flip ($\#O = 1$), etc.
- *Dynamic faults*: These are FPs that can only be sensitized by performing more than one operation *sequentially*: $\#O > 1$. Depending on $\#O$, a further classification can be made between *2-operation dynamic FPs* whereby $\#O = 2$, *3-operation dynamic FPs* whereby $\#O = 3$, etc.

The FFMs like Stuck-at Faults, Transition Faults [19], Read Destructive Faults [1], Coupling Faults, etc belong to static faults; the whole space of such faults can be found in [6, 20]. The space of dynamic faults will be defined in the next section.

3 Dynamic fault space

Dynamic faults can be divided into FPs describing single-cell dynamic faults (involving a single-cell), and FPs describing multi-cell dynamic faults (involving more than one cell). For multi-cell FPs, we restrict our analysis to two-cell FPs, because they are considered to be an important class for memory faults. Below single-cell dynamic FFMs and two-cell dynamic FFMs will be described.

3.1 Single-cell dynamic FFMs

Single-cell dynamic FFMs consist of FPs sensitized by applying more than one operation to a single cell *sequentially*. We will restrict our analysis to 2-operation dynamic faults. As mentioned in Section 2, a particular FP is denoted as $< S/F/R >$.

S describes *sensitizing operation sequence*, which sensitizes a fault F in the cell. Since two operations are considered, there are 18 possible Ss. They are: $xwywz$, $rxrx$, $rxwy$, $xwyry$, where $x, y, z \in \{0, 1\}$ and 'r' denotes a read operation and 'w' denotes a write operation. E.g., $0w1r1$ denotes a write 1 operation applied to a cell whose initial state is 0; the write is followed immediately with a read 1 operation.

F describes the value of the *faulty (i.e., victim)* cell (*v-cell*); $F \in \{0, 1, \uparrow, \downarrow\}$, where $\uparrow$ ($\downarrow$) denotes an up (down) transition due to a certain sensitizing *operation*.

R describes the logical value which appears at the output of the memory if the sensitizing operation applied to the v-cell is a *read* operation: $R \in \{0, 1, -\}$. A '$-$' in R means that the output data is not applicable; e.g., if $S = 0w0w1$, then no data will appear at the memory output, and for that reason R is replaced by a '$-$'.

In the following, only '$S = xwyry$', which have been verified to cause dynamic faults (see Section 4), will be considered. Given that $S = xwyry$, $F \in \{0, 1, \uparrow, \downarrow\}$, and $R \in \{0, 1, -\}$, it can be verified that there are 12 possible FPs $< S/F/R >$. These FPs are compiled into a set of three FFMs; they are listed in Table 1 together with their FPs:

1. *Dynamic Read Destructive Fault (dRDF)*: a write followed immediately by a read operation performed on a cell changes the data in the cell, and returns an *incorrect* value on the output. The dRDF consists of four FPs. Here, the write can be a transition write as well as a non-transition write operation.
2. *Dynamic Deceptive Read Destructive Fault (dDRDF)*.
3. *Dynamic Incorrect Read Fault (dIRF)*.

Table 1. List of single-cell dynamic FFMs

FFM	Fault primitives
dRDF	$< 0w0r0/\uparrow/1 >, < 0w1r1/\downarrow/0 >,$ $< 1w0r0/\uparrow/1 >, < 1w1r1/\downarrow/0 >$
dDRDF	$< 0w0r0/\uparrow/0 >, < 0w1r1/\downarrow/1 >,$ $< 1w0r0/\uparrow/0 >, < 1w1r1/\downarrow/1 >$
dIRF	$< 0w0r0/0/1 >, < 0w1r1/1/0 >,$ $< 1w0r0/0/1 >, < 1w1r1/1/0 >$

3.2 Two-cell dynamic FFMs

Two-cell dynamic FFMs consist of FPs sensitized by applying more than one operation *sequentially* to two cells: the *aggressor (a-cell)* and the v-cell. The a-cell is the cell to which the sensitizing operation (or state) should be applied in order to sensitize the fault, while the v-cell is the cell where the fault appears. Depending on how many operations are applied to the a-cell and to the v-cell, and the order they are applied, four types of S can be distinguished:

1. S_{aa}: the two operations are applied to the a-cell.
2. S_{vv}: the two operations are applied to the v-cell.
3. S_{av}: the first operation is applied to the a-cell, and the second one to the v-cell.

4. S_{va}: the first operation is applied to the v-cell, and the second one to the a-cell.

Since two operations are considered, there are $18 \times 4 = 72$ possible Ss (each S can be: $xwywz$, $rxrx$, $rxwy$, $xwyry$, where $x, y, z \in \{0, 1\}$). It is clear that despite the restriction to 2-operation dynamic faults, the number of FPs is still high. This calls for setting further restrictions on the sequences considered. Only S_{aa} and S_{vv} will be considered from now on. Further, and as we did in the case of single-cell faults, the $S = xwyry$ will be considered.

Faults caused by S_{aa}

The FPs for two-cell dynamic faults, where the two operations are applied to the a-cell, are represented as $< S_{aa}/F/R >=< S_a; S_v/F/R >_{a,v}$; S_a describes $xwyry$, while S_v describes the *state* of v-cell. Note that S_v does not specify an operation applied to the v-cell, but the *state* of the v-cell, since S_{aa} is considered. Now it can be verified easily that there are 8 possible FPs: 4 denoted as $< xwyry; 0/\uparrow/- >$ and 4 denoted as $< xwyry; 1/\downarrow/- >$. The 8 possible FPs are compiled into one dynamic FFM, referred to as *Dynamic Disturb Coupling Fault (dCFds)*: a write operation followed immediately by a read operation performed on the a-cell causes the v-cell to flip. The FPs of dCFds are given in the first block of Table 2.

Table 2. List of two-cell dynamic FFMs

FFM	Fault primitives
dCFds	$< 0w0r0; 0/\uparrow/- >, < 0w1r1; 0/\uparrow/- >,$ $< 1w0r0; 0/\uparrow/- >, < 1w1r1; 0/\uparrow/- >,$ $< 0w0r0; 1/\downarrow/- >, < 0w1r1; 1/\downarrow/- >,$ $< 1w0r0; 1/\downarrow/- >, < 1w1r1; 1/\downarrow/- >$
dCFrd	$< x; 0w0r0/\uparrow/1 >, < x; 0w1r1/\downarrow/0 >,$ $< x; 1w0r0/\uparrow/1 >, < x; 1w1r1/\downarrow/0 >$
dCFdrd	$< x; 0w0r0/\uparrow/0 >, < x; 0w1r1/\downarrow/1 >,$ $< x; 1w0r0/\uparrow/0 >, < x; 1w1r1/\downarrow/1 >$
dCFir	$< x; 0w0r0/0/1 >, < x; 0w1r1/1/0 >,$ $< x; 1w0r0/0/1 >, < x; 1w1r1/1/0 >$

Faults caused by S_{vv}

The FPs for two-cell dynamic faults, where the two operations are applied to the v-cell, are represented as $< S_{vv}/F/R >=< S_a; S_v/F/R >_{a,v}$; S_a describes the *state* of the a-cell, while S_v describes $xwyry$. Note that S_a does not specify an operation applied to the a-cell, but the *state* of the a-cell, since only S_{vv} is considered. Now it can be verified easily that there are 24 possible FPs: 12 denoted as $< 0; S_v/F/R >_{a,v}$, and 12 as $< 1; S_v/F/R >_{a,v}$; note that in both presentations, $< S_v/F/R >$ represents the 12 possible single-cell dynamic FPs discussed in Section 3.1. The 24 possible FPs are compiled into three dynamic FFMs; they are given together with their FPs in Table 2; in the table 'x' denotes 0 or 1.

1. *Dynamic Read Destructive Coupling Fault (dCFrd)*: a write followed immediately by a read operation performed on the v-cell changes the data in the v-cell and returns an *incorrect* value on the output, if the a-cell is in a given state. The dCFrd consists of eight FPs.
2. *Dynamic Deceptive Read Destructive Coupling Fault (dCFdrd).*
3. *Dynamic Incorrect Read Coupling Fault (dCFir).*

4 Importance of dynamic faults

In order to show the importance of dynamic faults, fault analysis based on defect injection and SPICE simulation has been performed. The defects are injected in the reduced electrical model of a DRAM, causing for example a (partial) open connections. This section describes the performed analysis and the acquired results [3].

Opens represent unwanted resistances on a signal line that is supposed to conduct perfectly. In this section, the simulation results of the open within a DRAM cell OC shown in Figure 1 will be discussed; for other examples see [3]. The behavior of the DRAM is studied after injecting and simulating OC. The analysis considers open resistances within the range ($10\Omega \leq R_{op} \leq 10$ MΩ) on a logarithmic scale using 5 points per decade, in addition to $R_{op} = \infty$ Ω. Each injected open in the memory model creates floating nodes, the voltage of which is varied between V_{DD} and GND on a linear scale using 10 points.

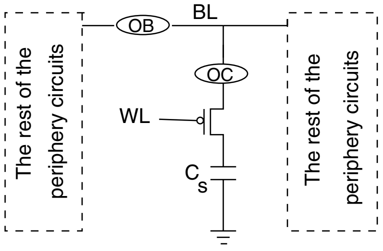

Figure 1. The opens OC and OB.

For each value of the open resistance (R_{op}) and of the initial floating node voltage (U_{init}), the following operation sequences are performed and inspected for proper functionality: $0w0$, $0w1$, $1w0$, $1w1$, $0r0$, $1r1$, $0w0r0$, $0w1r1$, $1w0r0$ and $1w1r1$ (e.g., $0w1$ denotes a write 1 applied to a cell whose content is 0). As a result, the faulty behavior resulting from the analysis of opens is represented as regions in the (U_{init}, R_{op}) plane. Each region contains a number of sensitized FPs that describe the FFM of the memory in this region.

As an example, the results of the fault analysis performed on OC are given in Figure 2, which shows the observed faulty behavior in the (U_{init}, R_{op}) plane. The figure shows a number of different fault regions for different combinations of U_{init} and R_{op}. In the figure, the static and dynamic sensitized faults are listed:

- Static Faults [20]:
  - $\text{TF}_d = <1w0/1/->$: a down transition fault.
  - $\text{TF}_u = <0w1/0/->$: an up transition fault.
  - $\text{IRF}_0 = <r0/0/1>$: incorrect read 0 fault.
  - $\text{RDF}_0 = <r0/\uparrow/1>$: read 0 destructive fault.
  - $\text{WDF}_0 = <w0/\uparrow/->$: write 0 disturb fault.
- Dynamic faults (see Table 1):
  - $\text{dRDF}_{00} = <0w0r0/\uparrow/1>$.
  - $\text{dRDF}_{10} = <1w0r0/\uparrow/1>$.
  - $\text{dDRDF}_{00} = <0w0r0/\uparrow/0>$.
  - $\text{dIRF}_{00} = <0w0r0/0/1>$.

The fault regions may be classified according to the initial floating node voltage under which they can be detected as follows.

A. Faults detectable with $U_{init} = V_{DD}$: They consist of fault regions A1 and A2.

B. Faults detectable with U_{init} = GND: They consist of fault regions B1, B2, B3, B4 and B5.

C. Faults only detectable with GND $< U_{init} < V_{DD}$: They consist of fault regions C1 and C2.

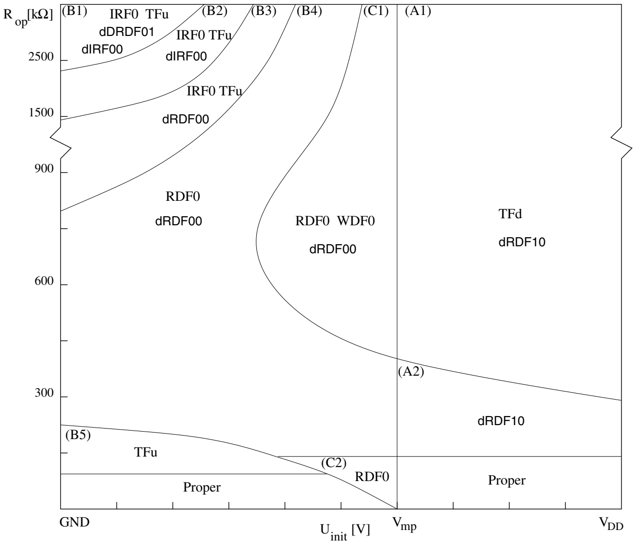

Figure 2. Analysis results for OC

Inspecting the faulty behavior shown in the figure reveals that, as a result of this cell open alone, almost all simulated dynamic sequences fail. This takes place in separate fault regions with their own initial voltage and defect resistance values. The fault region A2 only contains dRDF_{10}, which means that $1w0r0$ is the only failing S in this region. This, in turn, means that performing the traditional *static* analysis on this fault region reveals no improper memory behavior. Only by applying a *dynamic* sensitizing operation sequence is it possible to detect this improper behavior. This shows the significance of performing the dynamic analysis on memory devices.

A similar analysis has been done for the open OB (see Figure 1). The results show that some fault regions a dynamic fault behavior takes place in the absence of a static fault behavior [3]. Table 3 shows which dynamic sequences fail for OC and OB [3]. The table shows that for OC most, but not all, dynamic sequences fail. However, for OB all simulated dynamic sequences cause a fault.

Table 3. Observed dynamic FFMs for OC/ OB

Open	Dynamic faulty behavior
OC	All FPs of dRDF (see Table 1)
	Two dDRDF FPs: $<0w1r1/\downarrow/1>, <1w0r0/\uparrow/0>$
	Two dIRF FPs: $<0w0r0/0/1>, <1w1r1/1/0>$
OB	All FPs of Table 1

5 Testing static and dynamic faults

In Section 4 the existence of dynamic faults has been validated based on SPICE simulation. In Section 4, a systematic way to develop fault models for dynamic faults has been introduced. In this section, the conventional memory tests will be analyzed for their capability of detecting dynamic faults. However, first the march notation will be introduced.

5.1 March notation

A complete march test is delimited by the '{...}' bracket pair, while a march element is delimited by the '(...)' bracket pair. 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 an increasing ($\Uparrow$) or a decreasing ($\Downarrow$) address order. When the address order is not relevant, the symbol $\Updownarrow$ is be used.

5.2 Effectiveness of march tests

Table 4 gives the fault coverage of several well-known march tests regarding the proposed dynamic FFMs. The included march tests in the table are:

- MATS+ [13]: $\{\Updownarrow(w0); \Uparrow(r0,w1); \Downarrow(r1,w0)\}$
- March C- [11, 19]: $\{\Updownarrow(w0); \Uparrow(r0,w1); \Uparrow(r1,w0); \Downarrow(r0,w1); \Downarrow(r1,w0); \Updownarrow(r0)\}$
- March B [15]: $\{\Updownarrow(w0); \Uparrow(r0,w1,r1,w0,r0,w1); \Uparrow(r1,w0,w1); \Downarrow(r1,w0,w1,w0); \Downarrow(r0,w1,w0)\}$
- PMOVI [5]: $\{\Downarrow(w0); \Uparrow(r0,w1,r1); \Uparrow(r1,w0,r0); \Downarrow(r0,w1,r1); \Downarrow(r1,w0,r0)\}$

Table 4. Dynamic fault coverage for different march tests

FFM	March Tests							
	MATS+ ($5n$)	March C- ($10n$)	March B ($17n$)	PMOVI ($13n$)	March U ($13n$)	March SR (14n)	March LA (22n)	March LR (14n)
dRDF	0%	0%	50%	50%	50%	50%	50%	50%
dDRDF	0%	0%	0%	50%	0%	0%	50%	0%
dIRF	0%	0%	50%	50%	50%	50%	50%	50%
dCFds$_{xw\overline{x}r\overline{x}}$	0%	0%	50%	87.5%	50%	50%	100%	50%
dCFds$_{xwxrx}$	0%	0%	0%	0%	0%	0%	0%	0%
dCFrd	0%	0%	25%	50%	25%	25%	50%	25%
dCFdrd	0%	0%	0%	37.5%	0%	0%	50%	0%
dCFir	0%	0%	25%	50%	25%	25%	50%	25%

- March U [18]: $\{\Updownarrow (w0); \Uparrow (r0, w1, r1, w0); \Uparrow (r0, w1); \Downarrow (r1, w0, r0, w1); \Downarrow (r1, w0)\}$
- March SR [6]: $\{\Downarrow (w0); \Uparrow (r0, w1, r1, w0); \Uparrow (r0, r0); \Uparrow (w1); \Downarrow (r1, w0, r0, w1); \Downarrow (r1, r1)\}$
- March LA [17]: $\{\Updownarrow (w0); \Uparrow (r0, w1, w0, w1, r1); \Uparrow (r1, w0, w1, w0, r0); \Downarrow (r0, w1, w0, w1, r1); \Downarrow (r1, w0, w1, w0, r0); \Downarrow (r0)\}$
- March LR [16]: $\{\Updownarrow (w0); \Downarrow (r0, w1); \Uparrow (r1, w0, r0, w1); \Uparrow (r1, w0); \Uparrow (r0, w1, r1, w0); \Uparrow (r0)\}$

In the table, e.g., 100% denotes that the test detects all FPs of the correspondent FFM. For example, March C- detects 0% of dRDF FPs, while PMOVI detects 50% of them. The dCFds is divided into two types: dCFds$_{xw\overline{x}r\overline{x}}$ whereby the fault is caused by a transition write operation followed by a read; and dCFds$_{xwxrx}$ whereby the fault is caused by a non-transition write operation followed by a read; ($x \in \{0, 1\}$).

The table shows clearly that conventional march tests constructed to detect the static faulty behavior of a specific defect, do not necessarily detect its dynamic faulty behavior. None of the march tests of the table can cover the considered dynamic faults. In addition, a relative comparison of the fault coverage of the tests shows that March LA and PMOVI are the tests with the best coverage. The fact the the conventional tests do no cover dynamic faults calls for the introduction of new test.

6 Tests for dynamic faults

Dynamic faults are divided into faults involving a single cell and faults involving two cells. March tests for each of these two subclasses will be introduced separately.

6.1 Test for single-cell dynamic FFMs

March RAW1 ('read-after-write') given in Figure 3 detects all single-cell dynamic FFMs of Table 1, which are based on 'read-after-write'.

$$\boxed{\begin{array}{ccccccccccc} \{ \Updownarrow (w0) & ; & \Updownarrow (w0,r0) & ; & \Updownarrow (r0) & ; & \Updownarrow (w1,r1) & ; & \Updownarrow (r1) & ; & \\ M_0 & & M_1 & & M_2 & & M_3 & & M_4 & & \\ & & \Updownarrow (w1,r1) & ; & \Updownarrow (r1) & ; & \Updownarrow (w0,r0) & ; & \Updownarrow (r0) & \} & \\ & & M_5 & & M_6 & & M_7 & & M_8 & & \end{array}}$$

Figure 3. March RAW1

March RAW1 has a test length of $13n$ including the initialization. Table 5 shows by which march elements (i.e., M_0 through M_8) of March RAW1, each FP belonging to each single-cell dynamic FFM, is sensitized and detected. In the table, 'S' stands for 'sensitizing', while 'D' stands for detecting. Note that the test performs at most 2 operations per march element, which is important to restrict the sensitized faults to 2-operation dynamic.

Table 5. March RAW1 fault coverage

FFM	FP	S	D
dRDF	$< 0w0r0/\uparrow/1 >$	M_1	$\mathrm{M}_1, \mathrm{M}_2$
	$< 0w1r1/\downarrow/0 >$	M_3	$\mathrm{M}_3, \mathrm{M}_4$
	$< 1w0r0/\uparrow/1 >$	M_7	$\mathrm{M}_7, \mathrm{M}_8$
	$< 1w1r1/\downarrow/1 >$	M_5	$\mathrm{M}_5, \mathrm{M}_6$
dDRDF	$< 0w0r0/\uparrow/0 >$	M_1	M_2
	$< 0w1r1/\downarrow/1 >$	M_3	M_4
	$< 1w0r0/\uparrow/0 >$	M_7	M_8
	$< 1w1r1/\downarrow/1 >$	M_5	M_6
dIRF	$< 0w0r0/0/1 >$	M_1	$\mathrm{M}_1, \mathrm{M}_2$
	$< 0w1r1/1/0 >$	M_3	$\mathrm{M}_3, \mathrm{M}_4$
	$< 1w0r0/0/1 >$	M_7	$\mathrm{M}_7, \mathrm{M}_8$
	$< 1w1r1/1/0 >$	M_5	$\mathrm{M}_5, \mathrm{M}_6$

6.2 Test for two-cell dynamic FFMs

A test to detect all two-cell dynamic FFMs of Table 2 is given in Figure 4, and called *March RAW*. March RAW has a test length of $26n$ including the initialization. Note that the test coverage is not limited to two-operation dynamic faults, but detects also some dynamic faults where $\#O > 2$. Table 6 shows the march elements of March RAW responsible for sensitizing and detecting each of the faults listed in Table 2. In the table, a distinction is made between two cases: a) the v-cell has a higher address than the a-cell (i.e., $v > a$), and b) the v-cell has a lower address than the a-cell

($v < a$). In addition, in each entry the notation Sensitization/Detection is used. E.g., the $< 1w1r1; 0/\uparrow/- >$ is sensitized by M_4 and detected by M_5 when $v > a$; and it is sensitized by M_2 and detected by M_3 when $v < a$. Note that dCFrd and dCFir are not included in the table; the reader can easily verify that these faults are also covered by RAW, and any test detecting dCFdrd will also detect dCFrd and dCFir. Note additionally that March RAW also detects all FFM1s of Table 1; therefore only March RAW can be used to *detect* all dynamic faults introduced in this paper. If the purpose is to *diagnose* the FFM1s, then March RAW1 has to be used.

$$\boxed{\begin{array}{l} \{\ \underset{M_0}{\Updownarrow(w0)}\ ; \\ \underset{M_1}{\Uparrow(r0,w0,r0,r0,w1,r1)}\ ;\ \underset{M_2}{\Uparrow(r1,w1,r1,r1,w0,r0)}\ ; \\ \underset{M_3}{\Downarrow(r0,w0,r0,r0,w1,r1)}\ ;\ \underset{M_4}{\Downarrow(r1,w1,r1,r1,w0,r0)}\ ; \\ \underset{M_5}{\Updownarrow(r0)} \end{array}}$$

Figure 4. March RAW

Table 6. March RAW fault coverage

FFM	FP	$v > a$	$v < a$
dCFds	$< 0w0r0; 0/\uparrow/- >$	M_1/M_1	M_3/M_3
	$< 0w1r1; 0/\uparrow/- >$	M_1/M_1	M_3/M_3
	$< 1w1r1; 0/\uparrow/- >$	M_4/M_5	M_2/M_3
	$< 1w0r0; 0/\uparrow/- >$	M_4/M_5	M_2/M_3
	$< 0w0r0; 1/\downarrow/- >$	M_3/M_4	M_1/M_2
	$< 0w1r1; 1/\downarrow/- >$	M_3/M_4	M_1/M_2
	$< 1w1r1; 1/\downarrow/- >$	M_2/M_2	M_4/M_4
	$< 1w0r0; 1/\downarrow/- >$	M_2/M_2	M_4/M_4
dCFdrd	$< 0; 0w0r0/\uparrow/0 >$	M_3/M_3	M_1/M_1
	$< 0; 0w1r1/\downarrow/1 >$	M_3/M_4	M_1/M_2
	$< 0; 1w1r1/\downarrow/1 >$	M_2/M_2	M_4/M_4
	$< 0; 1w0r0/\uparrow/0 >$	M_2/M_3	M_4/M_5
	$< 1; 0w0r0/\uparrow/0 >$	M_1/M_1	M_3/M_3
	$< 1; 0w1r1/\downarrow/1 >$	M_1/M_2	M_3/M_4
	$< 1; 1w1r1/\downarrow/1 >$	M_4/M_4	M_2/M_2
	$< 1; 1w0r0/\uparrow/0 >$	M_4/M_5	M_2/M_3

7 Conclusions

In this paper the difference between static and dynamic faults in memories has been discussed; dynamic faults can take place in the absence of static faults as has been shown experimentally. An analytical approach for establishing fault models for dynamic faults has been presented. In addition, a set of fault models (under certain restrictions) has been introduced. The evaluation of the existing conventional memory tests shows that these tests cannot cover these fault models. Therefore, two new memory tests have been derived to target such specific faults.

The fault models introduced for dynamic faults have been restricted to faults sensitized by 'a write followed immediately with a read'. These faults have been observed while simulating DRAMs. The question that arises now is whether other sequential operations can also sensitize dynamic faults; e.g., 'a write followed immediately by a write', 'a read followed immediately by a read', etc. The widely used 'hammer tests' (i.e., repeat a write or a read operation sequentially) may indicate the existence of such dynamic faults. Furthermore, the 'Holey Shmoo problem' [7] in which the L1 cache of IBM System/390 G6 microprocessor fails to pass consecutive write patterns also indicate that dynamic faults can be caused by 'a write followed immediately by another write'. It is clear from the above, that the set of fault models for dynamic faults has to be explored, and the appropriate test algorithms have to be established.

References

[1] R.D. Adams and E.S. Cooley, "Analysis of a Deceptive Read Destructive Memory Fault Model and Recommended Testing", *In Proc. IEEE North Atlantic Test Workshop*, 1996.

[2] Z. Al-Ars, Ad J. van de Goor, "Impact of Memory Cell Array Bridges on the Faulty Behavior in Embedded DRAMs", *In Records of ATS*, pp. 282-289 , 2000.

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

[4] 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.

[5] J.H. De Jonge and A.J. Smeulders, "Moving Inversions Test Pattern is Thorough, Yet Speedy", *In Comp. Design*, pp. 169-173, 1976.

[6] S. Hamdioui, A.J. van de Goor, "Experimental Analysis of Spot Defects in SRAMs: Realistic Fault Models and Tests", *In Records of ATS*, pp. 131-138, Taipei, Taiwan, 2000.

[7] W. Huott, *et. al*, "The Attack of the 'Holey Shmoos': A Case of the Advanced DFD and Picosecond Imaging Circuit Analysis (PICA)", *In Proc. of IEEE International Test Conference,* pp. 883-891, 1999.

[8] 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.

[9] V.K. Kim and T. Chen, "On Comparing Functional Fault Coverage and Defect Coverage for Memory Testing", *IEEE Trans. on CAD*, V. 18, N. 11, pp. 1676-1683, 1999.

[10] 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, 1997.

[11] M. Marinescu, "Simple and Efficient Algorithms for Functional RAM Testing", *In Proc. of IEEE International Test Conference,* pp. 236-239, 1982.

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

[13] R. Nair, "An Optimal Algorithm for Testing Stuck-at Faults Random Access Memories", *IEEE trans. on Computers*, Vol. C-28, No. 3, pp. 258-261, 1979.

[14] C.A. Papachristou and N.B. Saghal, "An Improved Method for Detecting Functional Faults in Random Access Memories", *IEEE Trans. on Computers,* C-34, N. 3, pp. 110-116, 1985.

[15] 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.

[16] A.J. van de Goor and G. Gaydadjiev, "March LR: A Memory Test for Realistic Linked Faults", *In Proc. IEEE VLSI Test Symposium,* pp. 272-280, 1996.

[17] A.J. van de Goor *et al.*, "March LA: A Test for Linked Memory Faults," *in Proc. of European Design and Test Conf.*, 1999, p. 627.

[18] A.J. van de Goor and G.N. Gayadadjiev, "March U: A Test for All Unlinked Memory Faults, *IEE Proc. of Circuits Devices and Systems*, V. 144, No. 3, pp. 155-160, 1997.

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

[20] A.J. van de Goor and Z. Al-Ars, "Functional Fault Models: A Formal Notation and Taxonomy", *In Proc. of IEEE VLSI Test Symposium*, pp. 281-289, 2000.

[21] P.K. Veenstra, *et. al*, "Testing of Random Access Memories: Theory and Practice", *IEE Proc. of Circuits, Devices, and Systems*, **135**(1), pp. 24-28, 1988.