Linked Faults in Random Access Memories: Concept, Fault Models, Test Algorithms, and Industrial Results

Said Hamdioui, *Member, IEEE*, Zaid Al-Ars, *Member, IEEE*, Ad J. van de Goor, *Member, IEEE*, and Mike Rodgers, *Member, IEEE*

***Abstract*—The analysis of linked faults (LFs), which are faults that influence the behavior of each other, such that masking can occur, has proven to be a source for new memory tests, characterized by an increased fault coverage. However, many newly reported fault models have not been investigated from the point-of-view of LFs. This paper presents a complete analysis of LFs, based on the concept of fault primitives, such that the whole space of LFs is investigated and accounted for and validated. Some simulated defective circuits, showing linked-fault behavior, will be also presented. The paper establishes detection conditions along with new tests to detect each fault class. The tests are merged into a single test March SL detecting all considered LFs. Preliminary test results, based on Intel advanced caches, show that its fault coverage is high as compared with all other traditional tests and that it detects some unique faults; this makes March SL very attractive industrially.**

***Index Terms*—Fault coverage, functional fault models (FFMs), linked faults (LFs), march tests, memory testing.**

I. Introduction

SEMICONDUCTOR memories are an integral part of any modern ultralarge-scale integrated (ULSI) circuit. With the new generations of ULSIs, the memory will dominate the chip area (about 94% in 2014 [16]). Hence, their testing will become a major cost factor in the production of modern ULSIs. 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.

In order to account for the faulty behavior observed in memory devices, researchers introduced a number of fault models and fault modeling tools specific for memories. One interesting fault modeling tool is the concept of linked faults (LFs), which studies the effect fault models have on each other. Their importance has been validated experimentally by [20], which describes the results of testing 800 DRAM chips with different march tests. One march test (March LA) [19] designed specifically for detecting LFs, had a higher fault coverage than the other march tests. However, the design of March LA was based on the fault models which were known at that time. Experimental work, based on defect injection and circuit simulation for SRAMs and for DRAMs [3], [4], [6], [7], [11], has shown the existence of several new fault models.

This paper gives a precise definition of LFs, based on the new concept of fault primitives (FPs). It establishes the space of all possible LFs and divides it into different fault classes. Some simulated defective circuits showing linked faulty behavior will also be presented. The paper derives tests for each fault class; the tests are merged into a single test covering the whole space of LFs. This test is compared theoretically as well as industrially with other known tests.

The paper is organized as follows. Section II starts with defining the concept of FPs and fault models, and it thereafter classifies them and shows the scope of this paper. Section III discusses *simple* faults, while Section IV introduces the concept of *linked* faults. Section V establishes the space of LFs and its validation. Section VI develops the tests. Section VII presents a theoretical and an industrial evaluation of the newly introduced tests. Section VIII ends with the conclusion.

II. Memory Fault Classification

Faults in memory systems are divided into faults in memory-cell array, faults in address decoders, and faults in read/write circuits [14], [22]. In this paper, we will restrict ourself to only memory-cell array faults, since they are the dominant faults in memory systems.

This section first introduces the concept of a FP that will be used thereafter to classify the memory faults; the scope of the paper will also be given.

A. FPs

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: 1) a list of performed memory operations and 2) 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 and the read

Manuscript received February 4, 2003; revised April 23, 2003. This paper was recommended by Associate Editor K. Chakrabarty.

S. Hamdioui was with the Intel Corporation, Santa Clara, CA 95052 USA. He is now with the Computer Engineering Laboratory, Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: S.Hamdioui@ewi.tudelft.nl).

Z. Al-Ars and A. J. van de Goor are with the Computer Engineering Laboratory, Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, 2628 CD Delft, The Netherlands.

M. Rodgers is with the Intel Corporation, Santa Clara, CA 95052 USA.

Digital Object Identifier 10.1109/TCAD.2004.826578

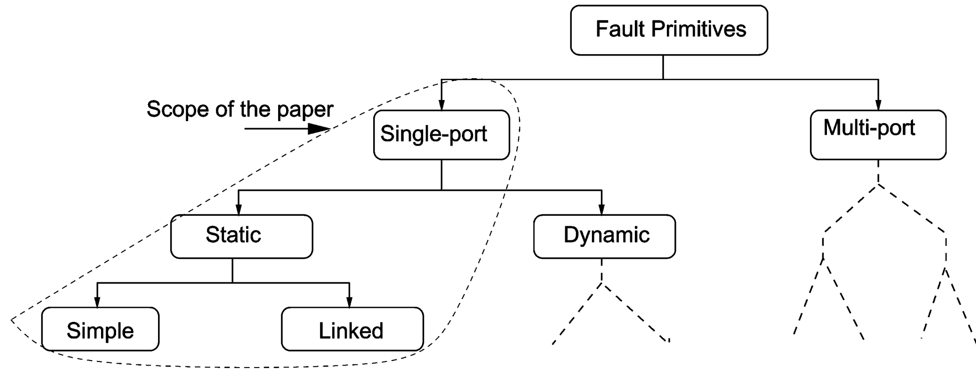


Fig. 1. Fault-primitive classification.

results (R) of S, in case S contains a read operation applied to the faulty cell. The combination of S, F, and R for a given memory failure is called a FP [18], and is denoted as $\langle S/F/R \rangle$. 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 contains a read operation applied to the faulty cell.

The concept of FPs allows for establishing a complete framework of all 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) as it has to be understood for memory devices [18]: an FFM is a nonempty set of FPs.

B. Classification

Fig. 1 shows the different classifications of the FPs. They can be classified based on:

1) the number of simultaneous operations required in S, into single-port and multiport faults.
2) the number of sequential operations required in S, into static and dynamic faults.
3) the way the FPs manifest themselves, into simple and LFs.

It is important to note that the three ways of classifying FPs are independent, since their definition is based on independent factors of S (see Fig. 1). For example, a dynamic FP can be single-port or multiport, simple or static.

1) Single-Port Versus Multiport Faults: Let $\#P$ define the number of ports required simultaneously to apply a S. For example, if a single read operation applied to cell c_1 causes that cell to flip, then $\#P = 1$; if two simultaneous read operations applied to the cell c_1 cause that cell to flip, then $\#P = 2$. Depending on $\#P$, FPs can be divided into single-port faults, and multiport faults.

- *Single-port faults*: These are FPs that require at most one port in order to be sensitized; that is $\#P \leq 1$. Note that single-port faults can be sensitized in single-port memories as well as in multiport memories.
- *Multiport faults:* These are FPs that can be only sensitized by performing two or more simultaneous operations via the different ports. For example, two-port faults can only be sensitized by performing two simultaneous operations via two different ports. Testing multiport faults is more complicated than testing single-port faults; they require specific patterns [8], [13].

2) Static Versus Dynamic Faults: Let $\#O$ define the number of different operations performed sequentially in a 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; that is $\#O \leq 1$. 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; that is $\#O > 1$. Experimental analysis of SRAMs and DRAMs, based on defect injection and SPICE simulation, shows that dynamic faulty behavior can take place in the absence of static faults [4], [10]. This faulty behavior requires more than one operation in order to be sensitized. For example, a write operation followed immediately with a read operation jointly cause the cell to flip; however, if only a write or a read operation is performed, the cell will not flip. Such faults may not necessarily be detected by conventional tests, and therefore adequate fault models and tests for dynamic faults are required [10].

3) Simple Versus LFs: Depending on the way FPs manifest themselves, they can be divided into simple faults and LFs.

- *Simple faults*: These are faults which cannot influence the behavior of each other. That means that the behavior of a simple fault cannot change the behavior of another one; therefore masking cannot occur.
- *LFs*: These are faults that do influence the behavior of each other. That means that the behavior of a certain fault can change the behavior of another one such that masking can occur [15], [22]. Note that LFs consist of two or more simple faults. Due to masking, testing of LFs is more complicated than testing of simple faults.

In the rest of this paper, we will focus on single-port, static, LFs; they will be discussed after presenting single-port, static, simple faults (see Fig. 1). From here on, the term "linked fault" refers to a "single-port, static linked fault," while the term "simple fault" refers to a "single-port, static simple fault."

III. Simple Faults

Simple faults can be divided into single-cell FPs (FPs) and multicell FPs. Single-cell FPs are FPs involving a single cell, while multicell FPs are FPs involving more than one cell. For multicell FPs, we will restrict our analysis to two-cell FPs (i.e., two-coupling FPs), because they are the vast majority of observed multicell FPs in practice [2]–[4], [6], [7], [11]. Below, single-cell simple and two-cell simple faults will be described.

A. Single-Cell Simple Faults

Before listing the possible single-cell FPs, the to-be-used notation will be explained.

$\langle S/F/R\rangle$ (or $\langle S/F/R\rangle_v$): denotes an FP involving a single-cell; the cell (victim cell) used to sensitize a fault is the same cell as where the fault appears.

S describes the value/operation sensitizing the fault; $S \in \{0, 1, 0w0, 1w1, 0w1, 1w0, 0r0, 1r1\}$, whereby 0 (1) denotes a zero (one) value, $0w0(1w1)$ denotes a write 0 (1) operation to a cell which contains a 0 (1), $0w1(1w0)$ denotes an up (down) transition write operation, and $0r0$ $(1r1)$ denotes a read 0 (1) operation from a cell containing 0 (1).

F describes the value of the faulty cell (v-cell) due to a certain sensitizing operation; $F \in \{0, 1\}$.

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 S does not contain a read operation applied to the v-cell, such that the output data is not applicable; e.g., if $S = 1w0$, then no data will appear at the memory output, and therefore R is replaced by a "$-$."

Because all possible values of S, F, and R can be enumerated for single-cell simple faults, it is possible to list all detectable FPs using the notation $\langle S/F/R\rangle$. It can be verified easily that there are 12 possible FPs [18], which have also been shown (based on SPICE simulation) to exist in real designs [2], [4], [6], [7], [11]. These FPs are compiled into a set of six FFMs. They are listed in Table I together with their FPs:

1) *State Fault* (SF): A cell is said to have an SF if the logic value of the cell flips before it is accessed, even if no operation is performed on it. This fault is special in the sense that no operation is needed to sensitize it. Therefore, it only depends on the initial stored value in the cell. The SF consists of two FPs: $\langle 0/1/-\rangle$ and $\langle 1/0/-\rangle$.
2) *Transition Fault* (TF).
3) *Write Destructive Faults* (WDF).
4) *Read Destructive Fault* (RDF) [2].
5) *Deceptive Read Destructive Fault* (DRDF) [2].
6) *Incorrect Read Fault* (IRF).

Note the above FFMs describe all possible single cell simple faults. Two additional FFMs will be mentioned: stuck-at fault (SAF) and stuck-open fault (SOF) [6]. Both FFMs can be considered as a composite FFM. The SAF can be denoted as $\langle\forall/0/-\rangle$ and $\langle\forall/1/-\rangle$. $\forall$ symbolizes all possible sensitizing operations. Therefore, $S = \forall$ can be replaced by any operation that sensitizes the fault. This leads to the following equivalent definitions of the SAF FPs:

TABLE I
List of Single-Cell Simple FFMs

#	FFM	FP	FP description
1	SF	SF_0	$< 1/0/- >$
		SF_1	$< 0/1/- >$
2	TF	TF_0	$< 1w0/1/- >$
		TF_1	$< 0w1/0/- >$
3	WDF	WDF_0	$< 0w0/1/- >$
		WDF_1	$< 1w1/0/- >$
4	RDF	RDF_0	$< 0r0/1/1 >$
		RDF_1	$< 1r1/0/0 >$
5	DRDF	$DRDF_0$	$< 0r0/1/0 >$
		$DRDF_1$	$< 1r1/0/1 >$
6	IRF	IRF_0	$< 0r0/0/1 >$
		IRF_1	$< 1r1/1/0 >$

- $\langle\forall/0/-\rangle = \{\langle 1/0/-\rangle, \langle 0w1/0/-\rangle\}$
- $\langle\forall/1/-\rangle = \{\langle 0/1/-\rangle, \langle 1w0/1/-\rangle\}$

For example, if the faulty behavior of a cell is said to resemble $\langle\forall/0/-\rangle$, then the cell exhibits the faulty behavior of each of the two FPs $\{\langle 1/0/-\rangle, \langle 0w1/0/-\rangle\}$.

The SOF is defined to describe the faulty behavior due to an open word line in the memory. Traditionally, an open line in the memory is expected to result in a failing 0w1 or 1w0 operation. Furthermore, it may and may not, depending on the technology of the sense amplifier, result in an incorrect read value. Using the FPs, the SOF can be defined as:

$$\text{SOF} = \{\langle 0w1/0/-\rangle, \langle 1w0/1/-\rangle, \langle 0r0/0/1\rangle, \langle 1r1/1/0\rangle\} = \text{TF} \cup \text{IRF}$$

B. Two-Cell Simple Faults

Two-cell simple faults consist of FPs sensitized by performing at most one operation, while considering the effect two different cells have on each other; they are also known as coupling faults. Such FPs can be presented as $\langle S/F/R\rangle = \langle S_a; S_v/F/R\rangle_{a,v}$, where S_a and S_v are the sequences performed on the aggressor (a-cell) and the v-cell, respectively. 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; or alternatively, the operation applied to the v-cell will sensitize the fault in the v-cell, provided that the a-cell is in a given state. Note that in the notation $\langle S_a; S_v/F/R\rangle_{a,v}$, if S_a is an operation, then S_v should be a state; while if S_a is a state, then S_v can be a state or an operation. $S_a, S_v \in \{0, 1, 0w0, 1w1, 0w1, 1w0, 0r0, 1r1\}$. For example, $\langle S_a; S_v/F/R\rangle_{a,v} = \langle 0w1; 0/1/-\rangle_{a,v}$ means that applying a 0w1 operation to the a-cell ($S_a = 0w1$) causes the v-cell to flip from 0 to 1 ($S_v = 0$ and $F = 1$). Since the output data is not applicable, R is replaced with "$-$." There are 36 possible two-cell FPs [18] which have been shown to exist in real designs [3], [6], [7], [11]. These FPs are compiled into seven FFMs listed in Table II.

1) State coupling fault (CFst): Two cells are said to have a CFst if the v-cell is forced into a given logic state only if the a-cell is in a given state, without performing any operation on the v-cell or on the a-cell. The CFst consists of four FPs; e.g., $\langle 0; 0/1/-\rangle$: the v-cell will be forced to 1 if the a-cell is in the state 0.

2) *Disturb coupling fault (CFds)*: Two cells are said to have a CFds if an operation (read, transition or nontransition write) performed on the a-cell causes the v-cell to flip. The CFds consists of 12 FPs; e.g., $\mathrm{CFds}_{xwy;0} = \langle xwy; 0/1/-\rangle, \mathrm{CFds}_{xrx;1} = \langle xrx; 1/0/\rangle$; $x, y \in \{0,1\}$, w denotes a write operation, and r denotes a read operation.
3) *Transition coupling fault (CFtr).*
4) *Write Destructive coupling fault (CFwd).*
5) *Read Destructive coupling fault (CFrd).*
6) *Deceptive Read Destructive coupling fault (CFdr).*
7) *Incorrect Read coupling fault (CFir).*

Note that the above FFMs describe all possible two-cell simple faults. However, two additional FFMs will be mentioned for historical reasons: the idempotent coupling fault (CFid) and the inversion coupling fault (CFin). The CFid is a subset of the CFds and consists of FPs that are only based on transition write operation. The CFin 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; it can be denoted as $\langle xw\bar{x}; y/\bar{y}/-\rangle$. The CFin, which has an academical origin, has never been shown to occur in real designs [3], [4], [7], [11].

IV. Linked Faults (LFs) Concept

LFs describe an interesting type of faulty behavior that takes place when more than one FP is sensitized in a defective memory. The definition of an LF is as follows: $\mathrm{LF}_1 = \mathrm{FP}_1 \rightarrow \mathrm{FP}_2$. This means that linked fault 1 consists of FP_1 linked to FP_2 (i.e., FP_1 and FP_2 are the linking FPs). If the sensitizing operation sequence (S_1) of FP_1 is applied first, it sensitizes a fault in the v-cell, and when the S_2 of FP_2 is applied next, it also sensitizes a fault in the same v-cell, but with a fault effect opposite to that of the S_1 of FP_1. The net result is that the fault effect of FP_2 masks the fault effect of FP_1.

LFs are very important for memory testing, because they reduce the fault coverage of tests, unless those tests have been designed specifically to cope with LFs [19], [20]. In order to clarify the idea of LFs, this section starts with two examples discussing the conditions FPs need to satisfy in order to be linked. The examples are followed by the definition of LFs, which is used to divide the space of possible LFs into a number of different classes. However, first the march notation will be given since it will be used in the two examples.

A. March Notation

A march test consists of a finite sequence of march elements [17]. A march element is a finite sequence of operations applied to each cell in the memory before proceeding to the next cell, according to a specified address order; i.e., ascending ($\Uparrow$), descending ($\Downarrow$), or irrelevant ($\Updownarrow$).

A complete march test is delimited by the "$\{\cdots\}$" bracket pair; while a march element is delimited by the "$(\cdots)$" bracket pair. The march elements are separated by semicolons, and the operations within a march element are separated by commas. The MATS+ march test [1] $\{\Updownarrow(w0); \Uparrow(r0, w1); \Downarrow(r1, w0)\}$ consists of the march elements $\Updownarrow\ (w0), \Uparrow\ (r0, w1)$, and $\Downarrow (r1, w0)$.

TABLE II
List of Two-Cell Simple FFMs; $x, y \in \{0,1\}$

#	FFM	FP	FP description
1	CFst	$\mathrm{CFst}_{x;0}$	$< 0; 0/1/- >, < 1; 0/1/- >$
		$\mathrm{CFst}_{x;1}$	$< 0; 1/0/- >, < 1; 1/0/- >$
2	CFds	$\mathrm{CFds}_{xwy;0}$	$< xwy; 0/1/- >$
		$\mathrm{CFds}_{xwy;1}$	$< xwy; 1/0/- >$
		$\mathrm{CFds}_{xrx;0}$	$< 0r0; 0/1/- >, < 1r1; 0/1/- >$
		$\mathrm{CFds}_{xrx;1}$	$< 0r0; 1/0/- >, < 1r1; 1/0/- >$
3	CFtr	$\mathrm{CFtr}_{x;0}$	$< 0; 1w0/1/- >, < 1; 1w0/1/- >$
		$\mathrm{CFtr}_{x;1}$	$< 0; 0w1/0/- >, < 1; 0w1/0/- >$
4	CFwd	$\mathrm{CFwd}_{x;0}$	$< 0; 0w0/1/- >, < 1; 0w0/1/- >$
		$\mathrm{CFwd}_{x;1}$	$< 0; 1w1/0/- >, < 1; 1w1/0/- >$
5	CFrd	$\mathrm{CFrd}_{x;0}$	$< 0; 0r0/1/1 >, < 1; 0r0/1/1 >$
		$\mathrm{CFrd}_{x;1}$	$< 0; 1r1/0/0 >, < 1; 1r1/0/0 >$
6	CFdr	$\mathrm{CFdr}_{x;0}$	$< 0; 0r0/1/0 >, < 1; 0r0/1/0 >$
		$\mathrm{CFdr}_{x;1}$	$< 0; 1r1/0/1 >, < 1; 1r1/0/1 >$
7	CFir	$\mathrm{CFir}_{x;0}$	$< 0; 0r0/0/1 >, < 1; 0r0/0/1 >$
		$\mathrm{CFir}_{x;1}$	$< 0; 1r1/1/0 >, < 1; 1r1/1/0 >$

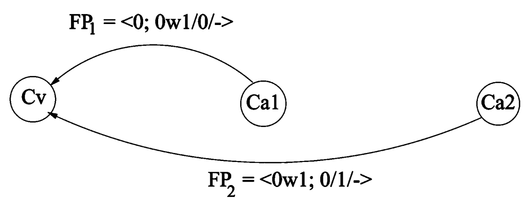


Fig. 2. $\mathrm{LF}_1 = \mathrm{FP}_1 \rightarrow \mathrm{FP}_2$.

B. Understanding LFs

Originally, faults would be called linked if a memory test (a march test, in particular) was capable of detecting the faults in isolation, yet the test may be unable to detect the faults when linked. The following two examples show how such a situation takes place.

Example 1: Consider $\mathrm{LF}_1 = \mathrm{FP}_1 \rightarrow \mathrm{FP}_2$ (see Fig. 2), whereby $\mathrm{FP}_1 = \langle 0; 0w1/0/-\rangle$ and $\mathrm{FP}_2 = \langle 0w1; 0/1/-\rangle$; note that FP_1 is a CFtr FP and FP_2 is a CFds FP; see Table II. Additionally, assume that the addresses of the a-cells C_{a1} and C_{a2} and of the v-cell C_v have the following relationship: $v < a1 < a2$. Then, MATS+ test $\{\Updownarrow(w0); \Uparrow(r0, w1); \Downarrow(r1, w0)\}$ cannot detect LF_1. This can be explained as follows: the first march element M_0 (i.e., $\Updownarrow(w0)$) sets all cells to 0. Next, when M_1 is applied to C_v it fails to set a 1 in that cell, because of FP_1. When M_1 is applied to C_{a2}, a 1 is written into C_{a2}, but also in C_v because of FP_2. Next, M_2 will not detect the fault because C_v contains a 1 (i.e., the fault effects mask each other). Note, however, that the test is able to detect both FP_1 and FP_2 when they are not linked (e.g., when FP_1 and FP_2 have different victims).

It is important to note that if FP_2 is sensitized first, then FP_1 will not be able to mask the fault effect of FP_2. This shows that if $\mathrm{FP}_1 \rightarrow \mathrm{FP}_2$ does not imply that $\mathrm{FP}_2 \rightarrow \mathrm{FP}_1$ (i.e., the linked to relationship between FPs is not commutative).

Example 2: Consider now $\mathrm{LF}_2 = \mathrm{FP}_1 \rightarrow \mathrm{FP}_2$ (see Fig. 3), whereby $\mathrm{FP}_1 = \langle 0w1; 0/1/-\rangle$ and $\mathrm{FP}_2 = \langle 0w1; 1/0/-\rangle$; note that both FP_1 and FP_2 are FPs of the CFds (see Table II).

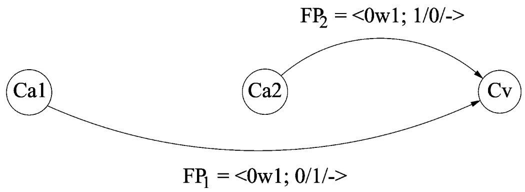


Fig. 3. LF_2: $\mathrm{FP}_1 \rightarrow \mathrm{FP}_2$.

Additionally, assume that the addresses of the a-cells C_{a1} and C_{a2} and of the v-cell C_v have the following relationship: $a1 < a2 < v$.

Then the well-known March C-test [12], [22]: $\{\Updownarrow (w0); \Uparrow (r0, w1); \Uparrow (r1, w0); \Downarrow (r0, w1); \Downarrow (r1, w0); \Updownarrow (r0)\}$ cannot detect LF_2. This can be explained as follows. When M_1 (i.e., $\Uparrow (r0, w1)$) is applied to C_{a1}, a 0 is first read, then 1 is written in that cell. Due to FP_1, 1 will be also written into C_v. Next, C_{a2} will be read within M_1, and then will be written with a 1. Due to FP_2, a 0 will be written into C_v; the fault is then masked. As a consequence, the read operation of M_1 applied to C_v will return the correct value; and the write operation will pass correctly. M_2 will also not detect the faults, since it does not contain an up transition write operation, required to sensitize FP_1 as well as FP_2. M_3 will first read 0 from C_v and then set it to 1. Next, it will read C_{a2} and set it to 1. However, due to FP_2, the C_v cell will flip to 0. Further, M_3 will read C_{a1} and set it to 1. Due to FP_1, the C_v cell will now flip to 1, which is the correct value that the C_v cell should contain. M_4 will, therefore, read the correct value from the v-cell. The fault LF_2 is thus not detected. Example 2 shows clearly that, even with a widely used test such as March C-, LFs will be not detected.

C. LFs Definition

Next, a definition of LFs, based on single-cell and two-cell FPs, will be given. If we assume that the faulty behavior of a memory contains two FPs that share the same v-cell, then $\mathrm{FP}_1 = \langle S_1/F_1/R_1\rangle$ is said to be linked to $\mathrm{FP}_2 = \langle S_2/F_2/R_2\rangle$ (denoted as $\mathrm{FP}_1 \rightarrow \mathrm{FP}_2$) if the following three conditions are satisfied.

1) **Read operations of $\mathbf{FP}_1$ and $\mathbf{FP}_2$ do not detect a fault**. This condition guarantees that both FP_1 and FP_2 are not detectable by read operations that S_1 or S_2 may contain. This is because if one of the FPs is detected, then it will make no sense to talk about a LFs, since the fault is already detected. For example, $\mathrm{RDF}_1 = \langle 1r1/0/0\rangle$ (see Table I) cannot be linked to any other FP since this fault is immediately detected on the output.
2) **$\mathbf{FP}_2$ masks $\mathbf{FP}_1$**. This means that $F_2 = \overline{F_1}$. This condition ensures that the faulty behavior of FP_2 hides the faulty behavior sensitized by FP_1 by masking it. For example, in the example of Fig. 3, sensitizing FP_1 by performing $S_1 = 0w1$, then sensitizing FP_2 by performing $S_2 = 0w1$, results in setting C_v to $F_1 = 1$ first, and then to $F_2 = \overline{F_1} = 0$, thereby masking the faulty behavior.
3) **$\mathbf{FP}_2$ is compatible with $\mathbf{FP}_1$**. This condition applies only in the case S_2 of the FP_2 is applied immediately after S_1 to the the same cell as the a-cell or the v-cell of FP_1. In that case, the final state of the a-cell (or of the v-cell) after performing S_1, should be the same as the initial state required by S_2 of FP_2. Condition 3 ensures that FP_2 can be sensitized after sensitizing FP_1. For example, $\mathrm{FP}_2 = \langle 1r1/0/0\rangle$ cannot be sensitized after $\mathrm{FP}_1 = \langle 0w1/0/-\rangle$, since after FP_1 the state of the v-cell is a faulty 0 which does not allow performing $1r1$ to sensitize FP_2.

V. LFs Space

This section determines the space of LFs. The classification of LFs will be given; each fault class will be discussed separately. Thereafter, some simulated defective circuits showing linked faulty behavior will be presented. However, first, the linking faults that do not require any operation to be sensitized will be presented.

A. Linking SF's and CFst's

SF's and CFst's are faults sensitized by an initialization part only (i.e., no operation part is required by their Ss). This makes them behave in a special way. An analysis of the behavior of these faults reveals two different cases [9], [21]:

1) SF's and CFst's can be linked among themselves. By assuming that the result of the fault effect is deterministic, these LFs will be simplified to a simple SF or CFst, which are detectable by the known march tests. If the result of the fault effect is not deterministic, then the faults can only be detected probabilistically by march tests, since the read operation may return random values.
2) SF's (or CFst's) can be linked to any other FP, and can other FPs be linked to SF's (or CFst's). SF's and CFst's are sensitized by the states of the a-cell, whereas the other faults (e.g., RDFs, CFds's) are sensitized by operations. Therefore, SF's and CFst's dominate the other single-cell and two-cell faults, respectively. SF's and CFst's will be present as long as the a-cell contains the sensitizing value, regardless of the fact that the v-cell is influenced by other faults and, therefore, they are detectable by the known march tests.

Based on the above, the SF's and the CFst's will not be treated further in this paper.

B. Classification of LFs

LFs can be divided into single-cell LFs, which involve a single cell, and multicell LFs, which involve more than one cell. Since the considered LFs are based on a combination of two simple FPs, and the the vast majority of observed simple FPs in practice are either single-cell FPs or two-cell FPs [2]–[4], [6], [7], [11], the LFs can involve at most three cells (i.e., $\#C \leq 3$). Fig. 4 shows the three classes of the LFs, whereby $\#C \leq 3$. Because the "linked to" relationship is not commutative, the classification also takes the order in which the two FPs are sensitized into consideration.

1) **The LFs involving a single cell (LF1s)**: They are based on a combination of two single-cell FPs. Both linked FPs have the same a-cell, which is also the v-cell. The LF is sensitized by sensitizing the two FPs sequentially in time.

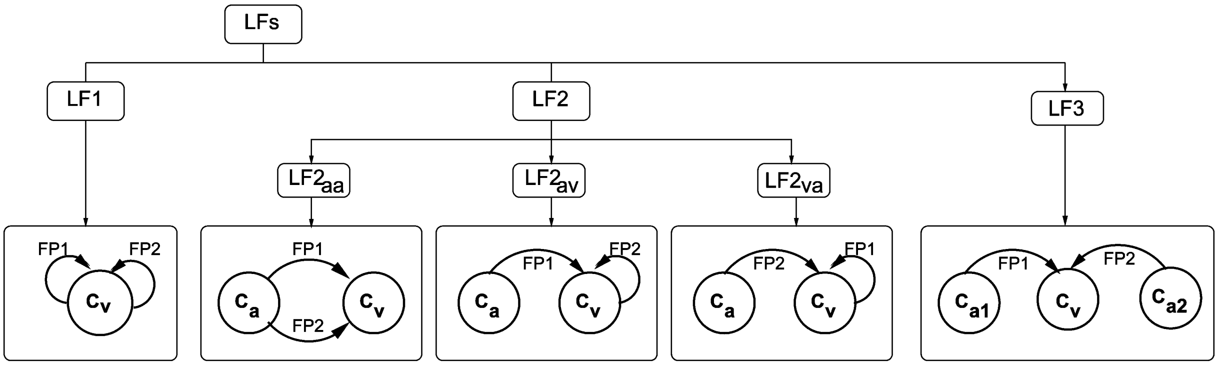


Fig. 4. Classification of LFs.

TABLE III
LF1s SPACE

FP_1	FP_2									
	TF_0	TF_1	WDF_0	WDF_1	RDF_0	RDF_1	IRF_0	IRF_1	DRDF_0	DRDF_1
TF_0: $<1w0/1/->$	$\overline{\mathrm{M}}$	$\overline{\mathrm{C}}$	$\overline{\mathrm{M}}$	$\mathbf{L}_1$	$\overline{\mathrm{M}}$	$\mathbf{L}_2$	$\overline{\mathrm{C}}$	$\overline{\mathrm{M}}$	$\overline{\mathrm{M}}$	D
TF_1: $<0w1/0/->$	$\overline{\mathrm{C}}$	$\overline{\mathrm{M}}$	$\mathbf{L}_3$	$\overline{\mathrm{M}}$	$\mathbf{L}_4$	$\overline{\mathrm{M}}$	$\overline{\mathrm{M}}$	$\overline{\mathrm{C}}$	D	$\overline{\mathrm{M}}$
WDF_0: $<0w0/1/->$	$\overline{\mathrm{M}}$	$\overline{\mathrm{C}}$	$\overline{\mathrm{M}}$	$\mathbf{L}_5$	$\overline{\mathrm{M}}$	$\mathbf{L}_6$	$\overline{\mathrm{C}}$	$\overline{\mathrm{M}}$	$\overline{\mathrm{M}}$	D
WDF_1: $<1w1/0/->$	$\overline{\mathrm{C}}$	$\overline{\mathrm{M}}$	$\mathbf{L}_7$	$\overline{\mathrm{M}}$	$\mathbf{L}_8$	$\overline{\mathrm{M}}$	$\overline{\mathrm{M}}$	$\overline{\mathrm{C}}$	D	$\overline{\mathrm{M}}$
RDF_0: $<0r0/1/1>$	D	D	D	D	D	D	D	D	D	D
RDF_1: $<1r1/0/0>$	D	D	D	D	D	D	D	D	D	D
IRF_0: $<0r0/0/1>$	D	D	D	D	D	D	D	D	D	D
IRF_1: $<1r1/1/0>$	D	D	D	D	D	D	D	D	D	D
DRDF_0: $<0r0/1/0>$	$\overline{\mathrm{M}}$	$\overline{\mathrm{C}}$	$\overline{\mathrm{M}}$	$\mathbf{L}_9$	$\overline{\mathrm{M}}$	$\mathbf{L}_{10}$	$\overline{\mathrm{C}}$	$\overline{\mathrm{M}}$	$\overline{\mathrm{M}}$	D
DRDF_1: $<1r1/0/1>$	$\overline{\mathrm{C}}$	$\overline{\mathrm{M}}$	$\mathbf{L}_{11}$	$\overline{\mathrm{M}}$	$\mathbf{L}_{12}$	$\overline{\mathrm{M}}$	$\overline{\mathrm{M}}$	$\overline{\mathrm{C}}$	D	$\overline{\mathrm{M}}$

2) **The LFs involving two cells (LF2s)**: These are based on a combination of two two-cell FPs, or on a combination of a single-cell FP and a two-cell FP. They are therefore divided into three types:
   a) The $\mathrm{LF2_{aa}}$: This LF is based on a combination of two two-cell FPs; both FPs have the same a-cell as well as the same v-cell.
   b) The $\mathrm{LF2_{av}}$: This LF is based on a combination of one two-cell FP_1 and one single-cell FP_2; whereby the two-cell FP is sensitized first.
   c) The $\mathrm{LF2_{va}}$: This LF is similar to $\mathrm{LF2_{av}}$. However, in this case, first the single-cell FP should be sensitized; thereafter the two-cell FP.
3) **The LFs involving three cells (LF3s)**: They are based on a combination of two two-cell FPs which have different a-cells, but the same v-cell.

In the remainder of this section, the domain of all possible LFs for LF1s–LF3s will be presented.

C. Single-Cell Faults

Table III shows all single-cell FPs (see also Table I) and the way they may be linked (i.e., whether $\mathrm{FP}_1 \rightarrow \mathrm{FP}_2$), with the exception of SF's, since they are not considered; each of the FPs can be any of those of Table I. The FPs are listed both horizontally (FP_1) and vertically (FP_2), and each potential $\mathrm{LF}_x = \mathrm{FP}_1 \rightarrow \mathrm{FP}_2$ is given an entry in the table, indicating if FP_1 can be (or not) linked to FP_2. There are four different entries used in the table:

TABLE IV
INSTANCES OF LF1s

#	LF1	#	LF1
L_1	$\mathrm{TF}_0 \rightarrow \mathrm{WDF}_1$	L_2	$\mathrm{TF}_0 \rightarrow \mathrm{RDF}_1$
L_3	$\mathrm{TF}_1 \rightarrow \mathrm{WDF}_0$	L_4	$\mathrm{TF}_1 \rightarrow \mathrm{RDF}_0$
L_5	$\mathrm{WDF}_0 \rightarrow \mathrm{WDF}_1$	L_6	$\mathrm{WDF}_0 \rightarrow \mathrm{RDF}_1$
L_7	$\mathrm{WDF}_1 \rightarrow \mathrm{WDF}_0$	L_8	$\mathrm{WDF}_1 \rightarrow \mathrm{RDF}_0$
L_9	$\mathrm{DRDF}_0 \rightarrow \mathrm{WDF}_1$	L_{10}	$\mathrm{DRDF}_0 \rightarrow \mathrm{RDF}_1$
L_{11}	$\mathrm{DRDF}_1 \rightarrow \mathrm{WDF}_0$	L_{12}	$\mathrm{DRDF}_1 \rightarrow \mathrm{RDF}_0$

1) **D**: This entry indicates that a fault is detected on the output by a read operation in FP_1 or FP_2. In other words, Condition 1 of the definition of LFs is not satisfied.
2) $\bar{\mathbf{M}}$: This entry indicates that FP_2 does not mask FP_1; i.e., Condition 2 is not satisfied.
3) $\bar{\mathbf{C}}$: This entry indicates that FP_2 is not compatible with FP_1; i.e., Condition 3 is not satisfied.
4) $\mathbf{L_x}$: This entry indicates that FP_1 can be *linked* to FP_2 since all conditions of the definition of LFs are satisfied.

The verification of $\mathrm{FP}_1 \rightarrow \mathrm{FP}_2$ is done in such way that the most intuitive condition to verify is checked first. If that condition is not satisfied, then it is listed in the corresponding entry in the table, irrespective of other linking conditions. For example, for FPs that are sensitized by a read operation, the detection condition (i.e., D) is checked first. Note that the entries in Table III are not symmetric with respect to the diagonal since the "*linked to*" relationship is not commutative.

As an example consider $\mathrm{L}_4 = \mathrm{TF}_1 \rightarrow \mathrm{RDF}_0$ in Table III, where an up-transition fault $(\langle 0w1/0/-\rangle)$ is linked to an RDF 0 $(\langle 0r0/1/1\rangle)$. This linked fault results from a failing transi-

TABLE V
INSTANCES OF $LF2_{aa}$ FAULTS

#	$LF2_{aa}$ ($FP_1 \rightarrow$CFds)	#	$LF2_{aa}$ ($FP_1 \rightarrow$CFwd)	#	$LF2_{aa}$ ($FP_1 \rightarrow$CFrd)
L_1	$CFds_{x_1O_1y_1;0} \rightarrow CFds_{x_2O_2y_2;1}$	L_9	$CFds_{xOy;0} \rightarrow CFwd_{y;1}$	L_{17}	$CFds_{xOy;0} \rightarrow CFrd_{y;1}$
L_2	$CFds_{x_1O_1y_1;1} \rightarrow CFds_{x_2O_2y_2;0}$	L_{10}	$CFds_{xOy;1} \rightarrow CFwd_{y;0}$	L_{18}	$CFds_{xOy;1} \rightarrow CFrd_{y;0}$
L_3	$CFtr_{x;0} \rightarrow CFds_{xOy;1}$	L_{11}	$CFtr_{x;0} \rightarrow CFwd_{x;1}$	L_{19}	$CFtr_{x;0} \rightarrow CFrd_{x;1}$
L_4	$CFtr_{x;1} \rightarrow CFds_{xOy;0}$	L_{12}	$CFtr_{x;1} \rightarrow CFwd_{x;0}$	L_{20}	$CFtr_{x;1} \rightarrow CFrd_{x;0}$
L_5	$CFwd_{x;0} \rightarrow CFds_{xOy;1}$	L_{13}	$CFwd_{x;0} \rightarrow CFwd_{x;1}$	L_{21}	$CFwd_{x;0} \rightarrow CFrd_{x;1}$
L_6	$CFwd_{x;1} \rightarrow CFds_{xOy;0}$	L_{14}	$CFwd_{x;1} \rightarrow CFwd_{x;0}$	L_{22}	$CFwd_{x;1} \rightarrow CFrd_{x;0}$
L_7	$CFdr_{x;0} \rightarrow CFds_{xOy;1}$	L_{15}	$CFdr_{x;0} \rightarrow CFwd_{x;1}$	L_{23}	$CFdr_{x;0} \rightarrow CFrd_{x;1}$
L_8	$CFdr_{x;1} \rightarrow CFds_{xOy;0}$	L_{16}	$CFdr_{x;1} \rightarrow CFwd_{x;0}$	L_{24}	$CFdr_{x;1} \rightarrow CFrd_{x;0}$

tion write operation $(0w1)$ followed by a read operation (r). External to the memory, where the internal voltages of the cells are not known, the write-read sequence is represented by $0w1r1$. However, since we know that the failing $0w1$ operation internally leaves the memory cell in state 0, the following read operation is in fact a $0r0$, instead of a $1r1$. The read operation fails to read the faulty 0 in the cell, sets the cell to 1 and results in 1 on the output. Although each of the individual operations $0w1$ and $0r0$ results internally in a fault, performing both operations as the external sequence $0w1r1$ does not result in any fault in the memory. From Table III one can conclude the following.

1) There are 12 LF1s, which are given in Table IV.
2) No $LF1 = FP_1 \rightarrow FP_2$ exists whereby $FP_1 \in \{RDF, IRF\}$ or $FP_2 \in \{TF, IRF, DRDF\}$.

D. Two-Cell Faults

The LF2s have been divided into three types (see Fig. 4). In the following, each of the three types will be discussed separately.

1) The $LF2_{aa}$ *Faults:* The $LF2_{aa}$ faults are LF2s, based on a combination of two two-cell FPs; both FPs have the same a-cell as well as the same v-cell. Thus the two FPs forming the $LF2_{aa}$ faults can be any of the FPs of Table II. The space of $LF2_{aa}$ faults has been analyzed in a similar way as for LF1s, and the results show the following [9].

1) There are 24 possible $LF2_{aa}$ faults given in Table V. A compact notation is used for FPs of Table II; e.g., $L_1, x_1O_1y_1$ can take on one of the following $\{0r0, 1r1, 0w0, 0w1, 1w0, 1w1\}$. Note that L_1 and L_2 require $x_2 = y_1$, since the compatibility condition has to be satisfied. E.g., $CFds_{x_1O_1y_1;0} \rightarrow CFds_{x_2O_2y_2;1}$ requires that $x_2 = y_1$ (i.e., the initial state required by $S_2 = x_2O_2y_2$ should be the same as the final state of the a-cell after performing $S_1 = x_1O_1y_1$). Note also that each $LF2_{aa}$ in Table V represents a set of two linked FPs; e.g., for L_1 and depending on $x_1O_1y_1$, the following 18 LFs can be distinguished:

$$\text{if } x_1O_1y_1 = 0r0, \quad \text{then } x_2O_2y_2 \in \{0r0, 0w1, 0w0\}$$
$$\text{if } x_1O_1y_1 = 0w0, \quad \text{then } x_2O_2y_2 \in \{0r0, 0w1, 0w0\}$$
$$\text{if } x_1O_1y_1 = 1w0, \quad \text{then } x_2O_2y_2 \in \{0r0, 0w1, 0w0\}$$
$$\text{if } x_1O_1y_1 = 1r1, \quad \text{then } x_2O_2y_2 \in \{1r1, 1w1, 1w0\}$$
$$\text{if } x_1O_1y_1 = 0w1, \quad \text{then } x_2O_2y_2 \in \{1r1, 1w1, 1w0\}$$
$$\text{if } x_1O_1y_1 = 1w1, \quad \text{then } x_2O_2y_2 \in \{1r1, 1w1, 1w0\}.$$

TABLE VI
INSTANCES OF $LF2_{av}$ FAULTS

#	$LF2_{av}$ ($FP_1 \rightarrow$WDF)	#	$LF2_{av}$ ($FP_1 \rightarrow$RDF)
L_1	$CFds_{xOy;0} \rightarrow WDF_1$	L_9	$CFds_{xOy;0} \rightarrow RDF_1$
L_2	$CFds_{xOy;1} \rightarrow WDF_0$	L_{10}	$CFds_{xOy;1} \rightarrow RDF_0$
L_3	$CFtr_{x;0} \rightarrow WDF_1$	L_{11}	$CFtr_{x;0} \rightarrow RDF_1$
L_4	$CFtr_{x;1} \rightarrow WDF_0$	L_{12}	$CFtr_{x;1} \rightarrow RDF_0$
L_5	$CFwd_{x;0} \rightarrow WDF_1$	L_{13}	$CFwd_{x;0} \rightarrow RDF_1$
L_6	$CFwd_{x;1} \rightarrow WDF_0$	L_{14}	$CFwd_{x;1} \rightarrow RDF_0$
L_7	$CFdr_{x;0} \rightarrow WDF_1$	L_{15}	$CFdr_{x;0} \rightarrow RDF_1$
L_8	$CFdr_{x;1} \rightarrow WDF_0$	L_{16}	$CFdr_{x;1} \rightarrow RDF_0$

2) No $LF_{aa} = FP_1 \rightarrow FP_2$ exists, whereby

$$FP_1 \in \{CFrd, CFir\} \quad \text{or} \quad FP_2 \in \{CFtr, CFir, CFdr\}.$$

2) The $LF2_{av}$ *Faults:* The $LF2_{av}$ faults are based on a combination of one two-cell FP_1 and one single-cell FP_2. In order to sensitize such a $LF2_{av}$, the two-cell FP_1 has to be sensitized first, and thereafter the single-cell FP_2. The FP_1 can be any FP of those of Table II; while FP_2 can be any FP of those of Table I. The space of $LF2_{av}$ faults has been analyzed in a similar way as for LF1s, and the results show the following [9].

1) There are 16 $LF2_{av}$ faults, given in Table VI.
2) No $LF_{av} = FP_1 \rightarrow FP_2$ exists, whereby

$$FP_1 \in \{CFrd, CFir\} \quad \text{or} \quad FP_2 \in \{TF, IRF, DRDF\}.$$

3) The $LF2_{va}$ *Faults:* The $LF2_{va}$ faults are based on a combination of one single-cell FP_1 and one two-cell FP_2. The FP_1 can be any FP of those of Table I, while FP_2 can be any FP of those of Table II. The space of $LF2_{va}$ faults has been analyzed and the results show that [9]:

1) There are 18 $LF2_{va}$ faults given in Table VII.
2) No $LF_{va} = FP_1 \rightarrow FP_2$ exists, whereby $FP_1 \in \{RDF, IRF\}$ or $FP_2 \in \{CFtr, CFir, CFdr\}$.

E. Three-Cell Faults

LF3s describe linking two two-cell FPs with different a-cells and the same v-cell; see Fig. 4. The instances of this class of LFs are exactly the same as those for two-cell $LF2_{aa}$ discussed in Section V-D1 and shown in Table V. The only difference is that for the instances of LF3, the compatibility condition does not apply since two different a-cells are used. For example $CFds_{x_1O_1y_1;1} \rightarrow CFds_{x_2O_2y_2;0}$ where $x_1Oy_1 = 0w1$ and

TABLE VII
INSTANCES OF $\mathrm{LF2_{va}}$ FAULTS.

$\mathrm{LF2}_{va}$ (TF→$\mathrm{FP_2}$)			
$\mathrm{LF_1}$	$\mathrm{TF_0} \rightarrow \mathrm{CFds}_{xOy;1}$	$\mathrm{LF_4}$	$\mathrm{TF_1} \rightarrow \mathrm{CFds}_{xOy;0}$
$\mathrm{LF_2}$	$\mathrm{TF_0} \rightarrow \mathrm{CFwd}_{x;1}$	$\mathrm{LF_5}$	$\mathrm{TF_1} \rightarrow \mathrm{CFwd}_{x;0}$
$\mathrm{LF_3}$	$\mathrm{TF_0} \rightarrow \mathrm{CFrd}_{x;1}$	$\mathrm{LF_6}$	$\mathrm{TF_1} \rightarrow \mathrm{CFrd}_{x;0}$
$\mathrm{LF2}_{va}$ (WDF→$\mathrm{FP_2}$)			
$\mathrm{LF_7}$	$\mathrm{WDF_0} \rightarrow \mathrm{CFds}_{xOy;1}$	$\mathrm{LF_{10}}$	$\mathrm{WDF_1} \rightarrow \mathrm{CFds}_{xOy;0}$
$\mathrm{LF_8}$	$\mathrm{WDF_0} \rightarrow \mathrm{CFwd}_{x;1}$	$\mathrm{LF_{11}}$	$\mathrm{WDF_1} \rightarrow \mathrm{CFwd}_{x;0}$
$\mathrm{LF_9}$	$\mathrm{WDF_0} \rightarrow \mathrm{CFrd}_{x;1}$	$\mathrm{LF_{12}}$	$\mathrm{WDF_1} \rightarrow \mathrm{CFrd}_{x;0}$
$\mathrm{LF2}_{va}$ (DRDF→$\mathrm{FP_2}$)			
$\mathrm{LF_{13}}$	$\mathrm{DRDF_0} \rightarrow \mathrm{CFds}_{xOy;1}$	$\mathrm{LF_{16}}$	$\mathrm{DRDF_1} \rightarrow \mathrm{CFds}_{xOy;0}$
$\mathrm{LF_{14}}$	$\mathrm{DRDF_0} \rightarrow \mathrm{CFwd}_{x;1}$	$\mathrm{LF_{17}}$	$\mathrm{DRDF_1} \rightarrow \mathrm{CFwd}_{x;0}$
$\mathrm{LF_{15}}$	$\mathrm{DRDF_0} \rightarrow \mathrm{CFrd}_{x;1}$	$\mathrm{LF_{18}}$	$\mathrm{DRDF_1} \rightarrow \mathrm{CFrd}_{x;0}$

$x_2Oy_2 = 0r0$ is a linked fault if LF3 is considered, but not if $\mathrm{LF2_{aa}}$ is considered.

F. Validation of the Linked Fault Space

In [3]–[4], [6]–[8], and [11], all possible resistive defects at the electrical level of the memory cell array for SRAMs and DRAMs have been defined, located, and simulated, using industrial designs. The simulation is based on a single-defect-at-a-time approach; i.e., only a single defect is simulated at a time. The found electrical faults have been transformed into FPs, and thereafter compiled into FFMs. All FPs considered for LFs in the previous section have been shown to exist. If a single-defect-at-a-time approach is considered for LFs, then it will be required that a single defect has to cause two different FPs with opposite fault effects. The set of LFs in that case will be empty for SRAMs and very small for DRAMs [9].

Since LFs are based on a combination of two FPs that have to be sensitized sequentially in time, the analysis of such faults has to consider a double-defect-at-a-time approach. This means that each defect can sensitize a FP which has an opposite fault effect compared with the other one, such that masking takes place.

Based on the "double-defect-at-a-time" approach, all introduced LFs in the previous section have to be considered realistic. If we assume that the defects are independent, then each of them can cause any FP, as has been shown in [3], [4], [6]–[8], and [11], and therefore all combinations satisfying the LF conditions are realistic. Note that the possible LFs caused by a single-defect are a subset of the LFs caused by two defects. Below, two examples (which have been simulated) of defective circuits show that linked faulty behavior will be present [9].

Example of LF1: An LF1 example, $\mathrm{TF_0} \rightarrow \mathrm{RDF_1}$, is shown in Fig. 5 for SRAMs. The $\mathrm{TF_0} = \langle 1w0/1/- \rangle$ is caused by the resistive defect D1, while the $\mathrm{RDF_1} = \langle 1r1/0/0 \rangle$ is caused by the resistive defect D2 in case the resistances of the two defects belong to certain ranges [2], [7], [8]. If now the sequence "w1, w0, r0" is applied to the cell. Then, the down transition write operation (1w0) will fail, and will be followed by a read operation (r), which will cause the cell to flip from 1 to 0; hence the linked fault $\mathrm{TF_0} \rightarrow \mathrm{RDF_1}$ will result. The linking process starts by performing $0w1$ on a cell, which fails to set the cell to 1. Then, performing the read operation fails to read the faulty 1 in the cell, sets the cell to 0 and results in 0 on the output. Although each of the individual operations $1w0$ and $1r1$ results internally in a fault, performing both operations as an external sequence does not result in any observable fault in the memory.

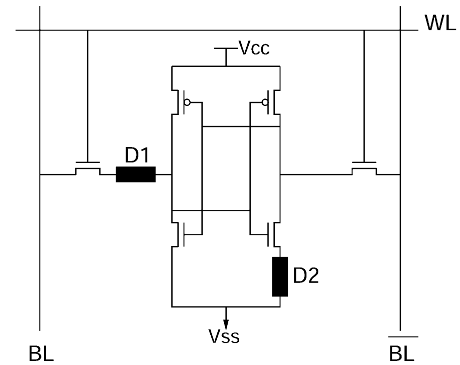


Fig. 5. Defects causing $\mathrm{TF_0} \rightarrow \mathrm{RDF_1}$.

Example of LF2: The LF2s are divided into three types (see Fig. 4), namely, $\mathrm{LF2_{aa}}, \mathrm{LF2_{av}}$, and $\mathrm{LF2_{va}}$. Fig. 6 shows an example of an $\mathrm{LF2_{av}}$ fault: $\mathrm{CFds}_{0w1;0} \rightarrow \mathrm{RDF_1}$ in SRAMs, where $\mathrm{CFds}_{0w1;0} = \langle 0w1; 0/1/ \rangle$ and $\mathrm{RDF_1} = \langle 1r1/0/0 \rangle$. Assume that the following sequence of operations will be performed in the following order:

1) Initialize the a-cell and the v-cell to 0.
2) Apply "r0, w1" to the a-cell.
3) Apply "r0, w1" to the v-cell.

First, the a-cell and the v-cell are initialized to 0. Next, the sequence "r0, w1" is performed to the a-cell: the read value will return a correct value and the w1 operation will operate correctly. However, due to $\mathrm{CFds}_{0w1;0} = \langle 0w1; 0/1/- \rangle_{a,v}$ caused by D1, the v-cell will flip to 1 [7], [11]. Next, the sequence "r0, w1" will be performed to the v-cell. Externally the r0 operation will return a correct value 0. However, since we know that the v-cell is flipped to 1, the performed operation internally is r1 and not r0. Due to $\mathrm{RDF_1} = \langle 1r1/0/0 \rangle$ caused by D2, the read operation will return 0 and cause the cell to flip back to 0, as no fault will be detected. Although the write operation applied to the a-cell, and the read operation, applied to the v-cell, each internally result in a fault; externally they do not result in any fault.

VI. TESTS FOR LINKED FAULTS

This section presents tests for LFs. They have been divided into three classes: LF1s, LF2s, and LF3s (see Fig. 4). The tests for each class will be discussed separately, in such a way that first the test conditions will be established, which thereafter will be used to develop test algorithms.

A. Testing Single-Cell LFs

The set of LF1s consist in total of 12 faults (see Table IV). In the following, first the detection conditions for LF1s will be presented. Thereafter, they will be compiled into a march test.

1) Detection Conditions: Below, detection conditions are given for detecting the $\mathrm{LF1} = \mathrm{FP_1} \rightarrow \mathrm{FP_2}$ faults of Table IV, based on the idea that any LF1 is detectable when at least one of the linking FPs, forming LF1, can be sensitized and detected in isolation (i.e., without allowing the other FP to mask the fault).

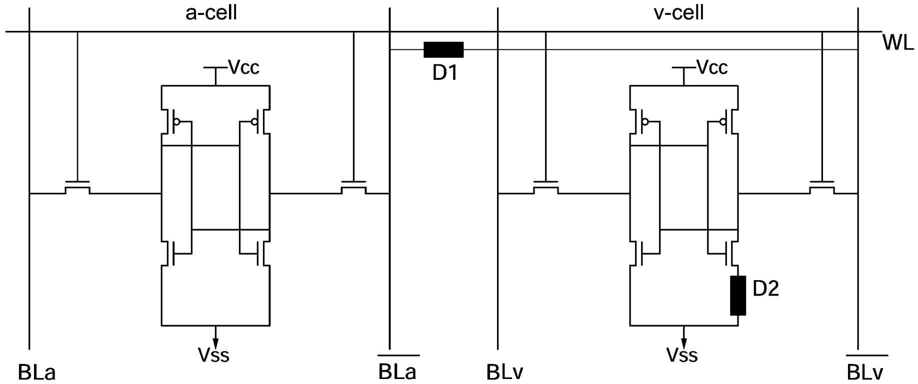


Fig. 6. Defects causing $\mathrm{LF2_{av}}: \mathrm{CFds}_{0w1;0} \rightarrow \mathrm{RDF}_1$.

a) Detection Condition for $\mathrm{FP}_1 \rightarrow \mathrm{WDF}_1$*:* The LF1, $\mathrm{FP}_1 \rightarrow \mathrm{WDF}_1$, whereby $\mathrm{FP}_1 \in \{\mathrm{TF}_0, \mathrm{WDF}_0, \mathrm{DRDF}_0\}$ (i.e., $\mathrm{L}_1, \mathrm{L}_5$, and L_9 in Table IV) are detectable if the WDF_1 is sensitized and detected in isolation. That is, any test detecting WDF_1 in isolation (if possible), will detect $\mathrm{L}_1, \mathrm{L}_5$, and L_9.

For the *simple* $\mathrm{WDF}_1 = \langle 1w1/0/- \rangle$, one can easily develop the detection condition a test has to satisfy in order to detect this fault. The fault is detectable by a march test which contains the three march elements: $\Updownarrow (w1); \Updownarrow (w1); \Updownarrow (r1, \ldots)$, where the three march elements can be merged into one or two elements.

The fault is first sensitized by applying a non transition "w1" operation (i.e., applying a "w1" to the cell which contains 1), and detected by reading the same cell. Since the sensitizing operation is 1w1, one has to make sure that the content of the v-cell is 0 before the first w1 operation. Otherwise, the fault may not be detected. If before performing the first "w1" operation, the content's cell was 1, then the first "w1" will be a nontransition write operation and therefore will cause the cell to flip to 0 due to $\mathrm{WDF}_1 = \langle 1w1/0/- \rangle$, while the second "w1" will be an up transition write and will put the correct 1 in the cell. The "r1" will then return a correct value, and the fault is therefore not detected. Thus one has to guarantee the the content of the v-cell is 0 before the first w1 operation; this can be done by adding a march element $\Updownarrow (\ldots, O0)$ to the begin of the test; the "O" denotes any operation, and "$O0$" guarantees that the content of the accessed cell is 0 before the first "w1" operation is applied. Now, the first write operation will be an up transition and the second "w1" operation will be a nontransition "w1" operation, which is required to sensitize the fault. The fault will be then detected by the "r1" operation, which will return a wrong value 0, rather than the expected value 1. Thus, a simple WDF_1 is detectable by a march test which contains the four march elements: $\Updownarrow (\ldots, O0); \Updownarrow (w1); \Updownarrow (w1); \Updownarrow (r1, \ldots)$, where the four march elements can be merged into one, two, or three elements.

For a *linking* WDF_1, however, the above condition is not sufficient, due to the masking. The "$O0$" in the condition can be:

1) a transition write operation (1w0): this can sensitize a TF_0 that can be linked to WDF_1 (see Table IV). Masking will take place and the fault will be not detected.
2) a nontransition write operation (0w0): this can sensitize a WDF_0 that can be linked to WDF_1. Masking will take place and the fault will be not detected.
3) a read operation (0r0): this can sensitize a DRDF_0 that can be linked to WDF_1. The fault will be then not detected due to the masking.

Only by allowing "$O0$" to be "$r0$," will the $\mathrm{TF}_0 \rightarrow \mathrm{WDF}_1$ and the $\mathrm{WDF}_0 \rightarrow \mathrm{WDF}_1$ be detected by detecting WDF_1 in isolation, since "$r0$" will not sensitize TF_0 nor WDF_0. However, $\mathrm{DRDF}_0 \rightarrow \mathrm{WDF}_1$ will not be detected since the "$r0$" can sensitize a DRDF_0 that can be linked to WDF_1. Adding an extra "$r0$" after $\Updownarrow (\ldots, r0)$ in the detection condition will detect $\mathrm{DRDF}_0 \rightarrow \mathrm{WDF}_1$ by detecting the DRDF_0 in isolation. The test condition detecting all $\mathrm{FP}_1 \rightarrow \mathrm{WDF}_1$, where $\mathrm{FP}_1 \in \{\mathrm{TF}_0, \mathrm{WDF}_0, \mathrm{DRDF}_0\}$ will then be the following.

<u>*Condition*</u> <u>$l\mathrm{WDF}_1$</u> (l in lWDF stands for linked): Any $\mathrm{FP}_1 \rightarrow \mathrm{WDF}_1$, whereby $\mathrm{FP}_1 \in \{\mathrm{TF}_0, \mathrm{WDF}_0, \mathrm{DRDF}_0\}$, is detectable by a march test which contains the following five march elements in the given order:

$$\Updownarrow (\ldots, r0); \quad \Updownarrow (r0); \quad \Updownarrow (w1); \quad \Updownarrow (w1); \quad \Updownarrow (r1, \ldots).$$

The five march elements can be merged into one, two, three or four march elements.

The $\mathrm{DRDF}_0 \rightarrow \mathrm{WDF}_1$ is detected by detecting DRDF_0, which is sensitized by the first march element (i.e., $\Updownarrow (\ldots, r0)$), and detected by the second one. The $\mathrm{TF}_0 \rightarrow \mathrm{WDF}_1$ and the $\mathrm{WDF}_0 \rightarrow \mathrm{WDF}_1$ are detected by detecting WDF_1, which is sensitized by the third followed by the fourth march element, and detected by the fifth one.

b) Detection Condition for $\mathrm{FP}_1 \rightarrow \mathrm{RDF}_1$*:* The LF1s, $\mathrm{FP}_1 \rightarrow \mathrm{RDF}_1$, whereby $\mathrm{FP}_1 \in \{\mathrm{TF}_0, \mathrm{WDF}_0, \mathrm{DRDF}_0\}$ (i.e., $\mathrm{L}_2, \mathrm{L}_6$, and L_{10} in Table IV) are detectable if the RDF_0 is sensitized and detected in isolation.

<u>*Condition*</u> <u>$l\mathrm{RDF}_1$</u>: The RDF_1 is detectable in isolation by a march test if the test contains the march element: $\Updownarrow (\ldots, r1, \ldots)$. Applying a "r1" operation will flip the cell to 0 and return the incorrect value 0.

Any operation performed before "1r1" operation of the march element, can be a "r1," a transition "w1" operation ("0w1"), or a non transition write-operation ("1w1"). By inspecting FP_1 in $\mathrm{FP}_1 \rightarrow \mathrm{RDF}_1$ (see Table IV), one can see that there is no FP that can be linked to RDF_1, and sensitized by "1r1", "0w1", or "1w1." Therefore, there is no possible masking, and $\Updownarrow (\ldots, r1, \ldots)$ will detect all $\mathrm{FP}_1 \rightarrow \mathrm{RDF}_1$ of Table IV.

c) Detection Condition for All LF1s: In a similar way as above, detection conditions for FPs linked to WDF_0 and to

RDF_0 (see Table IV) have been developed. All these detection conditions have been merged into the following condition [9]:
Condition LF1: Any LF1 of Table IV is detectable by a march test which contains the five march elements of Case A and the five march elements of Case B. The ten march elements can be merged into one to nine elements.

- Case A: to detect $\mathrm{L}_1, \mathrm{L}_2, \mathrm{L}_5, \mathrm{L}_6, \mathrm{L}_9$, and L_{10}

$$\Updownarrow(\ldots, r0);\quad \Updownarrow(r0);\quad \Updownarrow(w1);\quad \Updownarrow(w1);\quad \Updownarrow(r1,\ldots).$$

- Case B: to detect $\mathrm{L}_3, \mathrm{L}_4, \mathrm{L}_7, \mathrm{L}_8, \mathrm{L}_{11}$, and L_{12}

$$\Updownarrow(\ldots, r1);\quad \Updownarrow(r1);\quad \Updownarrow(w0);\quad \Updownarrow(w0);\quad \Updownarrow(r0,\ldots).$$

2) Test for LF1s: The test detecting all LF1s is shown in Fig. 7, and referred as March LF1. It has a test length of $11n$ (n is the size of the memory), including the initialization. It can be verified easily that the test satisfies Condition LF1: the second march element (i.e., M_1) of the test contains the five march elements of Case A; while M_2 contains the five march elements of Case B. Note that the first-read operation of M_2 can be removed without having any impact on the fault coverage of the test; M_1 will then contain the first march element of Case B, and M_2 will contain the other four march elements of Case B. However, the symmetrical structure of the test is desirable because it facilitates its implementation. Note also that the three march elements of the test can be merged into one or two elements.

B. Testing LF2s

LF2s are divided into three types: $\mathrm{LF2}_{\mathrm{aa}}, \mathrm{LF2}_{\mathrm{av}}$, and $\mathrm{LF2}_{\mathrm{va}}$ (see Section V-B). In the following, for each type, the detection conditions, as well as the tests, will be established.

1) Testing $\mathrm{LF2}_{\mathrm{aa}}$ *Faults:* The set of $\mathrm{LF2}_{\mathrm{aa}}$ faults consist of 24 faults; see Table V. Below detection conditions for all 24 faults will be given. Again, the idea is to detect one of the FPs of $\mathrm{LF}_{\mathrm{aa}} = \mathrm{FP}_1 \rightarrow \mathrm{FP}_2$ in isolation. Thereafter, the detection conditions will be used to derive the test.

a) Detection Conditions for $\mathrm{LF2}_{\mathrm{aa}}$*:* The introduction of the detection conditions for the $\mathrm{LF2}_{\mathrm{aa}}$ faults will be divided into three parts. First, the detection condition for the $\mathrm{FP}_1 \rightarrow \mathrm{CFds}$ (listed in the second column of Table V) will be established; then for the $\mathrm{FP}_1 \rightarrow \mathrm{CFwd}$ (listed in the fourth column of Table V); and thereafter for the $\mathrm{FP}_1 \rightarrow \mathrm{CFrd}$ (listed in the sixth column of Table V).

Detection Condition for $\mathbf{FP}_1 \rightarrow$ CFds

The $\mathrm{FP}_1 \rightarrow \mathrm{CFds}$ consists of LFs that can be $\mathrm{FP}_1 \rightarrow \mathrm{CFds}_{xOy;1}$ (i.e., $\mathrm{L}_1, \mathrm{L}_3, \mathrm{L}_5$, and L_7) and $\mathrm{FP}_1 \rightarrow \mathrm{CFds}_{xOy;0}$ (i.e., $\mathrm{L}_2, \mathrm{L}_4, \mathrm{L}_6$, and L_8); see the second column in Table V. They are detectable if the $\mathrm{CFds}_{xOy;1}$ and the $\mathrm{CFds}_{xOy;0}$ are sensitized and detected in *isolation*. That means that the detection of CFds in isolation will detect all $\mathrm{L}_i; 1 \le i \le 8$.

To detect a *simple* $\mathrm{CFds} = \langle xOy; z/\bar{z}/-\rangle$, where $x, y, z \in \{0, 1\}$ and O can be a read or a write, one can easily develop a detection condition.

Let us start with developing a detection condition for a simple $\mathrm{CFds} = \langle xOy; 0/1/-\rangle$, when the v-cell has a higher address than the a-cell ($v > a$). Below detection conditions are given for all possible xOy. It should be noted that extra operations may be inserted anywhere in the conditions as long as the required sensitizing operations are not impacted. For example, the sensitization of $\langle 1r1; 0/1/-\rangle$ requires a 1r1 operation; the condition $\Uparrow(r0, w1, r1, \ldots)$ can be extended with extra 0r0 or 1w1 operations; e.g., $\Uparrow(r0, w1, w1, r1, \ldots)$.

$$\{\ \underset{M_0}{\Updownarrow(w0)}\ ;\ \underset{M_1}{\Updownarrow(r0, r0, w1, w1, r1)}\ ;\ \underset{M_2}{\Updownarrow(r1, r1, w0, w0, r0)}\ \}$$

Fig. 7. March LF1.

$\mathrm{CFds} = \langle xOy; 0/1/-\rangle$	Detection condition; $v > a$
1. $\langle 0w1; 0/1/-\rangle$:	$\Uparrow(r0, w1, \ldots)$
2. $\langle 1w0; 0/1/-\rangle$:	$\Uparrow(r0, w1, w0, \ldots)$
3. $\langle 0w0; 0/1/-\rangle$:	$\Uparrow(r0, w0, \ldots)$
4. $\langle 1w1; 0/1/-\rangle$:	$\Uparrow(r0, w1, w1, \ldots)$
5. $\langle 0r0; 0/1/-\rangle$:	$\Uparrow(r0, \ldots)$
6. $\langle 1r1; 0/1/-\rangle$:	$\Uparrow(r0, w1, r1, \ldots)$

The above subconditions can be merged into a single short-detection condition that has to be satisfied in order to detect $\mathrm{CFds} = \langle xOy; 0/1/-\rangle$ when $v > a$. By merging the subconditions in the following sequence: 5, 1, 4, 6, 2, and 3, while considering the possibility of inserting extra operations without impacting the sensitizing operations, the condition will be:
Condition A.1: when $v > a : \Uparrow(r0, w1, w1, r1, w0, w0, \ldots)$
Note that the sequence of two w0 operations is required in order to guarantee that the second operation is a nontransition w0 operation; as required by $\mathrm{FP3} = \langle 0w0; 0/1/-\rangle$.

The table below shows which of the operations of the march element of Condition A.1 are responsible for the sensitization and the detection of each of the six FPs of $\mathrm{CFds} = \langle xOy; 0/1/-\rangle$; these are given in bold font. For example a nontransition write 0 operation required to sensitize the FP3 (i.e., $\langle 0w0; 0/1/-\rangle$) is achieved by the last operation of the march element; the r0 operation will then detect the fault.

$\mathrm{CFds} = \langle xOy; 0/1/-\rangle$	Detection condition; $v > a$
1. $\langle 0w1; 0/1/-\rangle$:	$\Uparrow(\mathbf{r0}, \mathbf{w1}, w1, r1, w0, w0, \ldots)$
2. $\langle 1w0; 0/1/-\rangle$:	$\Uparrow(\mathbf{r0}, w1, w1, r1, \mathbf{w0}, w0, \ldots)$
3. $\langle 0w0; 0/1/-\rangle$:	$\Uparrow(\mathbf{r0}, w1, w1, r1, w0, \mathbf{w0}, \ldots)$
4. $\langle 1w1; 0/1/-\rangle$:	$\Uparrow(\mathbf{r0}, w1, \mathbf{w1}, r1, w0, w0, \ldots)$
5. $\langle 0r0; 0/1/-\rangle$:	$\Uparrow(\mathbf{r0}, w1, w1, r1, w0, w0, \ldots)$
6. $\langle 1r1; 0/1/-\rangle$:	$\Uparrow(\mathbf{r0}, w1, w1, \mathbf{r1}, w0, w0, \ldots)$

In a similar way, we can develop the detection condition for $\mathrm{CFds}_{xOy;0}$ when $v < a$ as well as for $\mathrm{CFds}_{xOy;1}$ when $v > a$ and when $v < a$. The result is given in the next detection condition.
Condition CFds: A march test which contains both pairs of march elements of Case A, and both pairs of march elements of Case B detects all CFds (x and y can be 0 or 1; while O can be a read or a write operation).

- Case A: to detect $\mathrm{CFds}_{xOy;0} = \langle xOy; 0/1/-\rangle$:
  A.1 If $v > a : \Uparrow(r0, w1, w1, r1, w0, w0, \ldots)$;
  A.2 If $v < a : \Downarrow(r0, w1, w1, r1, w0, w0, \ldots)$;
- Case B: to detect $\mathrm{CFds}_{xOy;1} = \langle xOy; 1/0/-\rangle$:
  B.1 If $v > a : \Uparrow(r1, w0, w0, r0, w1, w1, \ldots)$;
  B.2 If $v < a : \Downarrow(r1, w0, w0, r0, w1, w1, \ldots)$;

The condition can be explained as follows. Case A detects the CFds when the value of fault effect is 1 (i.e.,

$\mathrm{CFds}_{xOy;0} = \langle xOy; 0/1/-\rangle$). Case A.1 sensitizes and detects the $\mathrm{CFds}_{xOy;0}$ when the address of the v-cell is higher than the address of the a-cell ($v > a$). Case A.2 sensitizes and detects the $\mathrm{CFds}_{xOy;0}$ when $v < a$. Note that each march element contains all possible sensitizing operation sequences $xOy \in \{0w0, 0w1, 1w0, 1w1, 0r0, 1r1\}$ since CFds consists of 12 FPs.

On the other hand, Case B is required to detect the CFds when the value of the fault effect is 0 (i.e., $\mathrm{CFds}_{xOy;1} = \langle xOy; 1/0/-\rangle$). Case B.1 sensitizes and detects the fault when $v > a$, while Case B.2 sensitizes and detects the fault when $v < a$.

Consider now faults linked to CFds. In order to facilitate the understanding of their detection condition, three cases will be distinguished (see the second column in Table V):

a) $\mathrm{FP}_1 \rightarrow \mathrm{CFds}$ whereby $\mathrm{FP}_1 \in \{\mathrm{CFtr}, \mathrm{CFwd}\}$. Note that FP_1 requires the application of the sensitizing operation to the v-cell while the a-cell is put in a certain state.

b) $\mathrm{FP}_1 \rightarrow \mathrm{CFds}$ whereby $\mathrm{FP}_1 = \mathrm{CFdr}$.

c) $\mathrm{CFds} \rightarrow \mathrm{CFds}$; i.e., $\mathrm{L}_1, \mathrm{L}_2$ of the second column in Table V.

First, it will be shown that Condition CFds also detects the $\mathrm{FP}_1 \rightarrow \mathrm{CFds}$ of Case a and Case b, but does not detect $\mathrm{CFds} \rightarrow \mathrm{CFds}$ of Case c. Thereafter, the condition will be transformed into a strong one taking masking into consideration such that all faults linked to CFds will be detected.

Case a: To detect $\mathrm{FP}_1 \rightarrow \mathrm{CFds}$ by detecting CFds in isolation, the test condition has to guarantee that the $\mathrm{FP}_1 \in \{\mathrm{CFtr}, \mathrm{CFwd}\}$ (see Table II) is not sensitized first. The sensitization of FP_1 requires the application of an operation to the v-cell while the a-cell has to be in a certain state. Selecting the a-cell first, applying a sequence of operations to it in order to sensitize the CFds; and then reading the v-cell will detect $\mathrm{FP}_1 \rightarrow \mathrm{CFds}$ by detecting the CFds in isolation, such that masking can not occur.

Consider for example the march element of Case A.1; i.e., $\Uparrow (r0, w1, w1, r1, w0, w0, \ldots)$; this element will detect all $\mathrm{FP}_1 \rightarrow \mathrm{CFds}$ where $\mathrm{CFds}_{xOy;0} = \langle xOy; 0/1/-\rangle$ and $v > a$. First the a-cell is accessed to perform the sensitizing operation sequence. Next the v-cell is read; the "$r0$" operation will return 1 instead of the expected value 0 and, therefore, the linked fault is detected. It is important that by applying "$r0$" to the v-cell no masking can occur. In $\mathrm{FP}_1 \rightarrow \mathrm{CFds}$, where $\mathrm{FP}_1 \in \{\mathrm{CFtr}, \mathrm{CFwd}\}$, the sensitization of CFtr and CFwd requires write operations. Therefore, they cannot be sensitized with the "$r0$" operation.

In a similar way, one can show that Case B detects all $\mathrm{FP}_1 \rightarrow \mathrm{CFds}_{xOy;0}$ by detecting $\mathrm{CFds}_{xOy;0}$ in isolation. Case B.1 detects the fault when $v > a$, while Case B.2 detects the fault when $v < a$.

Case b: To detect $\mathrm{CFdr} \rightarrow \mathrm{CFds}$ by detecting CFds in isolation, the detection condition has to guarantee that the $\mathrm{CFdr} = \langle x; yry/\bar{y}/y\rangle$ (see Table II) is not sensitized first. The sensitization of CFdr requires the application of a read operation to the v-cell while the a-cell has to be in a certain state. Selecting the a-cell first, applying a sequence of operations to it in order to sensitize the CFds; and then reading the v-cell will detect $\mathrm{CFdr} \rightarrow \mathrm{CFds}$ by detecting the CFds in isolation; masking can not occur due to the nature of CFdr as will be shown in the next example.

Consider Case A.1 $= \Uparrow (r0, w1, w1, r1, w0, w0, \ldots)$; this element will detect all $\mathrm{CFdr} \rightarrow \mathrm{CFds}$ where $\mathrm{CFds}_{xOy;0} = \langle xOy; 0/1/-\rangle$, and $v > a$; for example it will detect: $\langle 0; 1r1/0/1\rangle \rightarrow \langle 1w0; 0/1/-\rangle$ as explained next. First, the a-cell is accessed to perform the sensitizing operation sequence "$r1, w0$." The v-cell flips then form 0 to 1. Next, the v-cell is read and the CFdr is sensitized (i.e., the v-cell will flip back to 0). However, and due to the nature of CFdr, the read operation will return to 1 instead of the expected value 0 and, therefore, the linked fault is detected.

In a similar way, one can show that Case B detects all $\mathrm{CFdr} \rightarrow \mathrm{CFds}_{xOy;0}$ by detecting $\mathrm{CFds}_{xOy;0}$ in isolation. Case B.1 detects the fault when $v > a$, while Case B.2 detects the fault when $v < a$.

Case c: To detect $\mathrm{CFds} \rightarrow \mathrm{CFds}$ (i.e., L_1 and L_2 in Table V), one has to detect one of the two CFds's in isolation. Since the two CFds's involve the same a-cell as well as the same v-cell, masking may occur when the operation sequences are applied to the a-cell in order to sensitize the two CFds's. For instance, consider the following linked fault (i.e., L_1 of Table V):

$$\mathrm{CFds}_{x_1O_1y_1;0} \rightarrow \mathrm{CFds}_{x_2O_2y_2;1}$$

where, for example,

$$\mathrm{CFds}_{x_1O_1y_1;0} = \langle 0w1; 0/1/-\rangle \quad \text{and}$$
$$\mathrm{CFds}_{x_2O_2y_2;1} = \langle 1w0; 1/0/-\rangle.$$

It will be shown next that condition CFds can not detect the above fault, and therefore needs to be transformed into a strong condition taking masking into consideration.

As it has been mentioned previously, condition CFds is

A.1 $\Uparrow (r0, w1, w1, r1, w0, w0, \ldots)$
A.2 $\Downarrow (r0, w1, w1, r1, w0, w0, \ldots)$
B.1 $\Uparrow (r1, w0, w0, r0, w1, w1, \ldots)$
B.2 $\Downarrow (r1, w0, w0, r0, w1, w1, \ldots)$

Case A.1 will *not* detect the fault $\mathrm{CFds}_{x_1O_1y_1;0} \rightarrow \mathrm{CFds}_{x_2O_2y_2;1}$ when $v > a$ due to masking: the application of $x_1O_1y_1 = 0w1$ to the a-cell will cause the v-cell to flip to 1, and the $x_2O_2y_2 = 1w0$ will cause the cell to re-flip to 0. The "$r0$" operation, when applied to the v-cell, will return the expected value and the fault is thus not detected. Case A.2 will also *not* detect the fault when $v < a$, also due to the masking; a similar explanation can be given.

Case B.1 will *not* detect the fault when $v > a$ also due to the masking: the application of $x_2O_2y_2 = 1w0$ to the a-cell will cause the v-cell to flip from 1 to 0, and the $x_1O_1y_1 = 0w1$ will cause the cell to re-flip to 1. The "$r1$" operation, when applied to the v-cell, will return the expected value and the fault is thus not detected. Case B.2 will also *not* detect the fault when $v > a$, also due to the masking; a similar explanation can be given.

Therefore, Condition CFds can not detect $\mathrm{CFds} \rightarrow \mathrm{CFds}$, due to the masking. However, the condition can be transformed into a strong one, taking the masking into consideration and therefore detecting the fault.

Consider again the fault L_1 of Table V

$$\mathrm{CFds}_{x_1 O_1 y_1;0} \rightarrow \mathrm{CFds}_{x_2 O_2 y_2;1}$$

where

$$\mathrm{CFds}_{x_1 O_1 y_1;0} = \langle 0w1; 0/1/-\rangle \quad \text{and}$$
$$\mathrm{CFds}_{x_2 O_2 y_2;1} = \langle 1w0; 1/0/-\rangle.$$

For $v > a$, it is impossible to detect the fault with the march elements of Case A.1, since masking will always occur. However, the march element can be modified and used to detect the fault, not when $v > a$, but when $v < a$. The modification can be done as follows.

1) Add "$O1$" at the end of the march element of Case A.1, such that the contents of the cells become 1 after they have been accessed (O can be read or write). That is, change Case A.1 into

$$\Uparrow (r0, w1, w1, r1, w0, w0, \ldots, O1).$$

2) When now the a-cell with $v < a$ is accessed, which occurs after the v-cell has been accessed because $v < a$ and the $\Uparrow$ addressing is used, the $\mathrm{CFds}_{x_1 O_1 y_1;0}$ will not be sensitized since the state of the v-cell is 1, while the $\mathrm{CFds}_{x_1 O_1 y_1;0}$ requires the v-cell to be 0. However, the $\mathrm{CFds}_{x_2 O_2 y_2;1}$ will be sensitized, and will cause the v-cell $(v < a)$ to flip to 0.
3) Adding the march element $\Uparrow (r1, \ldots)$ after the modified march element of Case A.1 will detect the fault, since the v-cell $(v < a)$ will be accessed first and the read operation will return the wrong value 0 instead of the expected one 1.
4) The $\mathrm{CFds}_{x_1 O_1 y_1;0} \rightarrow \mathrm{CFds}_{x_2 O_2 y_2;1}$ is, thus, detectable when $v < a$ with the two following march elements, performed in the given order

$$\Uparrow (r0, w1, w1, r1, w0, w0, \ldots, O1); \quad \Uparrow (r1, \ldots).$$

In a similar way, one can modify the march element of Case A.2 in order to detect the same linked fault when $v > a$. The fault will be detectable with the two following march elements, performed in the given order

$$\Downarrow (r0, w1, w1, r1, w0, w0, \ldots, O1); \quad \Downarrow (r1, \ldots).$$

The reader can easily verify that the new derived two pairs of march elements of Case A.1 and Case A.2 also detect the L_2 of Table V ($\mathrm{CFds}_{x_1 O_1 y_1;1} \rightarrow \mathrm{CFds}_{x_2 O_2 y_2;0}$) by detecting $\mathrm{CFds}_{x_1 O_1 y_1;1}$ in isolation. Therefore, they detect all $\mathrm{CFds} \rightarrow \mathrm{CFds}$.

Note that with only Cases A.1 and A.2, all $\mathrm{CFds} \rightarrow \mathrm{CFds}$ (i.e., L_1 and L_2 of Table V) will be detected. Thus, only two of the four march elements of condition CFds are needed to be modified in order to cover all $\mathrm{CFds} \rightarrow \mathrm{CFds}$; the results are summarized in the following condition [9]. Cases B.1 and B.2 are obtained in a similar way as for Case A.1 and Case A.2.

Condition $\mathrm{CFds} \rightarrow \mathrm{CFds}$: A march test, which contains both pairs of march elements of Case A, or both pairs of march elements of Case B, or both pairs of march elements of Case C, or both pairs of march elements of Case D, detects all $\mathrm{CFds} \rightarrow \mathrm{CFds}$.

- Case A:

A.1 $\Uparrow (r0, w1, w1, r1, w0, w0, \ldots, O1); \quad \Uparrow (r1, \ldots)$
A.2 $\Downarrow (r0, w1, w1, r1, w0, w0, \ldots, O1); \quad \Downarrow (r1, \ldots)$

- Case B:

B.1 $\Uparrow (r1, w0, w0, r0, w1, w1, \ldots, O0); \quad \Uparrow (r0, \ldots)$
B.2 $\Downarrow (r1, w0, w0, r0, w1, w1, \ldots, O0); \quad \Downarrow (r0, \ldots)$

- Case C:

C.1 $\Uparrow (r0, w1, w1, r1, w0, w0, \ldots, O1); \quad \Uparrow (r1, \ldots)$
C.2 $\Downarrow (r1, w0, w0, r0, w1, w1, \ldots, O0); \quad \Downarrow (r0, \ldots)$

- Case D:

D.1 $\Downarrow (r0, w1, w1, r1, w0, w0, \ldots, O1); \quad \Downarrow (r1, \ldots)$
D.2 $\Uparrow (r1, w0, w0, r0, w1, w1, \ldots, O0); \quad \Uparrow (r0, \ldots).$

It has been shown that condition CFds is required to detect all $\mathrm{FP}_1 \rightarrow \mathrm{CFds}$ whereby $\mathrm{FP}_1 \in \{\mathrm{CFtr}, \mathrm{CFwd}, \mathrm{CFdr}\}$, and that condition $\mathrm{CFds} \rightarrow \mathrm{CFds}$ is required to detect all $\mathrm{CFds} \rightarrow \mathrm{CFds}$. These two detection conditions can be merged into a single detection condition that has to be satisfied in order to detect all $\mathrm{LF2_{aa}}$ faults, $\mathrm{FP}_1 \rightarrow \mathrm{CFds}$, given in the second column of Table V. The condition is given below.

Condition lCFds: A march test which contains both pairs of march elements of Case A, or both pairs of march elements of Case B, or both pairs of march elements of Case C, or both pairs of march elements of Case D, detects all $\mathrm{LF2_{aa}}$ faults, $\mathrm{FP}_1 \rightarrow \mathrm{CFds}$, by detecting the CFds in isolation (Note: each march element takes on all possible operation sequences sensitizing a CFds: 0r0, 0w1, 1w1, 1r1, 1w0, and 0w0).

- Case A:

A.1

$$\Uparrow (r0, w1, w1, r1, w0, w0, \ldots, O1)$$
$$\Uparrow (r1, w0, w0, r0, w1, w1, \ldots)$$

A.2

$$\Downarrow (r0, w1, w1, r1, w0, w0, \ldots, O1)$$
$$\Downarrow (r1, w0, w0, r0, w1, w1, \ldots)$$

- Case B:

B.1

$$\Uparrow (r1, w0, w0, r0, w1, w1, \ldots, O0)$$
$$\Uparrow (r0, w1, w1, r1, w0, w0, \ldots)$$

B.2

$$\Downarrow (r1, w0, w0, r0, w1, w1, \ldots, O0)$$
$$\Downarrow (r0, w1, w1, r1, w0, w0, \ldots)$$

- Case C:

C.1

$$\Uparrow (r0, w1, w1, r1, w0, w0, \ldots, O1)$$
$$\Uparrow (r1, w0, w0, r0, w1, w1, \ldots)$$

C.2

$$\Downarrow (r1, w0, w0, r0, w1, w1, \ldots, O0)$$
$$\Downarrow (r0, w1, w1, r1, w0, w0, \ldots)$$

- Case D:

D.1

$$\Downarrow (r0, w1, w1, r1, w0, w0, \ldots, O1)$$
$$\Downarrow (r1, w0, w0, r0, w1, w1, \ldots)$$

D.2

$$\Uparrow (r1, w0, w0, r0, w1, w1, \ldots, O0)$$
$$\Uparrow (r0, w1, w1, r1, w0, w0, \ldots).$$

Detection Condition for $\mathbf{FP}_1 \rightarrow \mathrm{CFwd}$

The $\mathrm{FP}_1 \rightarrow \mathrm{CFwd}$ consists of the $\mathrm{FP}_1 \rightarrow \mathrm{CFwd}_{x;1}$ (i.e., $\mathrm{L}_9, \mathrm{L}_{11}$, L_{13}, and L_{15}) and the $\mathrm{FP}_1 \rightarrow \mathrm{CFwd}_{x;0}$ (i.e., $\mathrm{L}_{10}, \mathrm{L}_{12}, \mathrm{L}_{14}$, and L_{16}); see the fourth column in Table V. These faults are detectable if one of the FPs forming the LFs is sensitized and detected in isolation. For example the detection of $\mathrm{CFwd}_{x;1}$ and the $\mathrm{CFwd}_{x;0}$ in isolation will guarantee the detection of all $\mathrm{L}_i; 9 \leq i \leq 16$.

Consider a simple $\mathrm{CFwd}_{x;0} = \langle x; 0w0/1/-\rangle_{a,v}$. To detect this fault, the a-cell has to be set first to the state "x" ($x \in \{0,1\}$) and, thereafter, the v-cell has to be accessed with "0w0" to sensitize the fault, followed with "r0" to detect it. One possible detection condition is that the march test has to contain the following two pairs of march elements:

$$\text{if } v < a: \quad \Updownarrow (\ldots, Ox); \quad \Uparrow (\ldots, w0, w0, r0, \ldots)$$
$$\text{if } v > a: \quad \Updownarrow (\ldots, Ox); \quad \Downarrow (\ldots, w0, w0, r0, \ldots)$$

Where "Ox" denotes "rx" or "wx;" $x \in \{0,1\}$. The first march element of each pair will set all cells (including the a-cell) to the state "x." The second march element accesses first the v-cell to sensitize and detect the $\mathrm{CFwd}_{x;0}$, before the a-cell is accessed. Since the sensitizing operation is "0w0," one has to make sure that the content of the v-cell is 1 before the first "$w0$" in $(\ldots, w0, w0, r0, \ldots)$ is applied; otherwise the fault may not be detected. For example, if the content of the v-cell is 0, before the first "$w0$" is applied, then the first write in $(\ldots, w0, w0, r0, \ldots)$ will be a nontransition write and therefore will flip the v-cell to 1 due to $\mathrm{CFwd}_{x;0} = \langle x; 0w0/1/-\rangle_{a,v}$, while the second write will become a down transition write and will mask the fault by writing 0 in the v-cell, such that reading the v-cell will return the correct value 0, and the fault is therefore not detected. Thus, one has to guarantee that the content of the v-cell is 1 before the two successive "w0" operations are applied. This can be done as follows:

$$\text{if } v < a: \quad \Updownarrow (\ldots, Ox); \quad \Uparrow (\ldots, O1, w0, w0, r0, \ldots)$$
$$\text{if } v > a: \quad \Updownarrow (\ldots, Ox); \quad \Downarrow (\ldots, O1, w0, w0, r0, \ldots).$$

The "$O1$" can be a "r1" operation, a transition "w1" operation, or a nontransition "w1" operation. For a linking $\mathrm{CFwd}_{x;0}$, however, the above condition is not yet sufficient due to the masking. The "$O1$" in the condition can sensitize FPs that can be linked to $\mathrm{CFwd}_{x;0}$ and, therefore, the fault will not be detected due to the masking (see also Table V).

1) If "$O1$" is a transition "w1" operation, the $\mathrm{CFtr}_{x,1}$ may be sensitized. The $\mathrm{CFtr}_{x,1}$ can be linked with $\mathrm{CFwd}_{x;0}$ (L_{12} of Table V), and the fault will be not detected. In that case, when the transition "w1 operation" is applied to v-cell, the "w1" will fail and the cell will remain in its state 0 due to $\mathrm{CFtr}_{x;1}$; as a consequence the first "w0" in the second march element will describe a nontransition "w0" operation that will cause the v-cell to flip to 1 due to $\mathrm{CFwd}_{x;0}$. When now the second 'w0' will be performed the cell will be put to 0 and the fault is then masked. The read operation will return a correct value, and the fault is not detected.
2) If "$O1$" is a nontransition "w1" operation, $\mathrm{CFwd}_{x,1}$ may be sensitized. The $\mathrm{CFwd}_{x,1}$ can be linked with $\mathrm{CFwd}_{x;0}$ (L_{14} of Table V), and the fault will be not detected.
3) If "$O1 = r1$," the $\mathrm{CFdr}_{x,1}$ may be sensitized. The $\mathrm{CFdr}_{x,1}$ can be linked with $\mathrm{CFwd}_{x;0}$ (L_{16} of Table V), and the fault will be not detected.

By allowing "$O1$" to be "$r1$," the $\mathrm{CFtr}_{x,1}$ and $\mathrm{CFwd}_{x,1}$ can not be sensitized and, therefore, the above detection condition will guarantee the detection of $\mathrm{CFtr}_{x;1} \rightarrow \mathrm{CFwd}_{x;0}$ and $\mathrm{CFwd}_{x;1} \rightarrow \mathrm{CFwd}_{x;0}$, as well as $\mathrm{CFds}_{yOx;1} \rightarrow \mathrm{CFwd}_{x;0}$ by detecting $\mathrm{CFwd}_{x;0}$ in isolation. However, this will *not* guarantee the detection of $\mathrm{CFdr}_{x;1} \rightarrow \mathrm{CFwd}_{x;0}$, because the "$r1$" can sensitize the $\mathrm{CFdr}_{x;1}$. Adding an extra "$r1$" after "$O1$" (where $O = r$) and before the first '$w0$' in $\Uparrow (\ldots, O1, w0, w0, r0, \ldots)$ will detect $\mathrm{CFdr}_{x;1} \rightarrow \mathrm{CFwd}_{x;0}$ by detecting the $\mathrm{CFdr}_{x;1}$ in isolation. The test condition for $\mathrm{FP}_1 \rightarrow \mathrm{CFwd}_{x;0}$, where $\mathrm{FP}_1 \in \{\mathrm{CFds}_{yOx;1}, \mathrm{CFtr}_{x;1}, \mathrm{CFwd}_{x;1}, \mathrm{CFdr}_{x;1}\}$ will then be the following:

$$\text{if } v < a: \quad \Updownarrow (\ldots, Ox); \quad \Uparrow (\ldots, r1, r1, w0, w0, r0, \ldots)$$
$$\text{if } v > a: \quad \Updownarrow (\ldots, Ox); \quad \Downarrow (\ldots, r1, r1, w0, w0, r0, \ldots).$$

where $x \in \{0,1\}$.

In a similar way, one can construct the detection condition for detecting $\mathrm{FP}_1 \rightarrow \mathrm{CFwd}_{x;1}$ (see the fourth column in Table V); the detection conditions will be

$$\text{if } v < a: \quad \Updownarrow (\ldots, Ox); \quad \Uparrow (\ldots, r0, r0, w1, w1, r1, \ldots)$$
$$\text{if } v > a: \quad \Updownarrow (\ldots, Ox); \quad \Downarrow (\ldots, r0, r0, w1, w1, r1, \ldots).$$

The two detection conditions together form the detection condition to detect all $\mathrm{FP}_1 \rightarrow \mathrm{CFwd}$ faults given in the fourth column of Table V. The detection condition is given below [9].

Condition lCFwd: A march test detects all $\mathrm{FP}_1 \rightarrow \mathrm{CFwd}_{x;1}$ (i.e., $\mathrm{L}_9, \mathrm{L}_{11}, \mathrm{L}_{13}$, and L_{15}) and all $\mathrm{FP}_1 \rightarrow \mathrm{CFwd}_{x;0}$ faults (i.e., $\mathrm{L}_{10}, \mathrm{L}_{12}, \mathrm{L}_{14}$, and L_{16}) if the test contains the two pairs of march elements of Case A and the two pairs of march elements of Case B; since $x \in \{0,1\}$ each pair of march elements describes two pairs.

- Case A: to detect $\mathrm{FP}_1 \rightarrow \mathrm{CFwd}_{x;0}$

A.1 if $v < a$:
$$\Updownarrow(\ldots, Ox); \quad \Uparrow(\ldots, r1, r1, w0, w0, r0, \ldots)$$
A.2 if $v > a$:
$$\Updownarrow(\ldots, Ox); \quad \Downarrow(\ldots, r1, r1, w0, w0, r0, \ldots)$$

- Case B: to detect $\mathrm{FP}_1 \rightarrow \mathrm{CFwd}_{x;1}$

B.1 if $v < a$:
$$\Updownarrow(\ldots, Ox); \quad \Uparrow(\ldots, r0, r0, w1, w1, r1, \ldots)$$
B.2 if $v > a$:
$$\Updownarrow(\ldots, Ox); \quad \Downarrow(\ldots, r0, r0, w1, w1, r1, \ldots).$$

Detection Condition for $\mathbf{FP}_1 \rightarrow$ CFrd

The $\mathrm{FP}_1 \rightarrow \mathrm{CFrd}$ consists of the $\mathrm{FP}_1 \rightarrow \mathrm{CFrd}_{x;1}$ (i.e., $\mathrm{L}_{17}, \mathrm{L}_{19}, \mathrm{L}_{21}$, and L_{23}) and the $\mathrm{FP}_1 \rightarrow \mathrm{CFrd}_{x;0}$ (i.e., $\mathrm{L}_{18}, \mathrm{L}_{20}, \mathrm{L}_{22}$, and L_{24}); see the sixth column of Table V. They are detectable if the $\mathrm{CFrd}_{x;1}$ and the $\mathrm{CFrd}_{x;0}$ are sensitized and detected in isolation. That means that the detection of CFrd in isolation will detect all $\mathrm{L}_i; 17 \leq i \leq 24$. To detect the $\mathrm{CFrd}_{x;1}$ and the $\mathrm{CFrd}_{x;0}$, the a-cell has to be set first to the state 'x', and thereafter the v-cell has to be accessed to sensitize and detect the fault. The two faults are detectable if the following condition is satisfied.

Condition lCFrd: A march test detects all CFrds in isolation if it contains the two pairs of march elements of Case A.1, and the two pairs of march elements of Case B.1 (since $x \in \{0, 1\}$ each pair of march elements describes two pairs).

- Case A: to detect $\mathrm{CFrd}_{x;0} = \langle x; r0/1/1 \rangle$

A.1 if $v < a$:$\Updownarrow(\ldots, Ox); \Uparrow(\ldots, r0, \ldots)$
A.2 if $v > a$:$\Updownarrow(\ldots, Ox); \Downarrow(\ldots, r0, \ldots)$

- Case B: to detect $\mathrm{CFrd}_{x;1} = \langle x; r1/0/0 \rangle$

B.1 if $v < a$:$\Updownarrow(\ldots, Ox); \Uparrow(\ldots, r1, \ldots)$
B.2 if $v > a$:$\Updownarrow(\ldots, Ox); \Downarrow(\ldots, r1, \ldots)$

The condition can be explained as follows. Case A will detect the CFrd in isolation when the value of the fault effect is 1 (i.e., $\mathrm{CFrd}_{x,0}$). Case A.1 detects CFrd when $v < a$, while Case A.2 detects the same fault when $v > a$. Case A.1 detects the fault as follows. The first march element will initialize all memory cells to "x" ($x \in \{0, 1\}$) including the a-cell. In the second march elements, the operation "$r0$" will sensitize and detect the fault in the v-cell, before the a-cell is accessed. Any operation performed before the "r0" operation within the same march element can be a "r0," a transition "w0" operation ("1w0"), or a non transition "w0" operation ("0w0"). By inspecting the sixth column in Table V, one can see that there is no FP that can be linked to $\mathrm{CFrd}_{x,0}$ and sensitized by "0r0", "1w0", or "0w0" applied to the v-cell. Therefore no masking is possible. A similar explanation applies to Case A.2.

Case B will detect the CFrd in isolation when the value of the fault effect is 0 (i.e., $\mathrm{CFrd}_{x,1}$). Case B.1 will detect the $\mathrm{CFrd}_{x,1}$ when $v < a$, while Case B.2 will detect the $\mathrm{CFrd}_{x,1}$ when $v > a$,

Note that Condition lCFrd is a subset of Condition lCFwd; therefore any test detecting $\mathrm{FP}_1 \rightarrow \mathrm{CFwd}$ also detects $\mathrm{FP}_1 \rightarrow \mathrm{CFrd}$.

b) Tests for $\mathrm{LF2}_{\mathrm{aa}}$*:* The test detecting all $\mathrm{LF2}_{\mathrm{aa}}$ faults is shown in Fig. 8, and referred to as *March LF2aa*. It has a test length of $41n$. In order to show that the test detects all $\mathrm{LF2}_{\mathrm{aa}}$ faults, it suffices to verify that the test satisfies condition lCFds and condition lCFwd. Condition lCFrd does not have to be verified since it is a subset of condition lCFwd.

$$\{ \underbrace{\Updownarrow(w0)}_{M_0};\ \underbrace{\Uparrow(r0,r0,w1,w1,r1,r1,w0,w0,r0,w1)}_{M_1};\ \underbrace{\Uparrow(r1,r1,w0,w0,r0,r0,w1,w1,r1,w0)}_{M_2};\ \underbrace{\Downarrow(r0,r0,w1,w1,r1,r1,w0,w0,r0,w1)}_{M_3};\ \underbrace{\Downarrow(r1,r1,w0,w0,r0,r0,w1,w1,r1,w0)}_{M_4} \}$$

Fig. 8. March $\mathrm{LF2}_{\mathrm{aa}}$.

Condition lCFds is satisfied as follows. M_1 and M_2 of the test contain the two march elements of Case A.1, while M_3 and M_4 contain the two march elements of Case A.2. Note that the last operation of $\mathrm{M}_1(\mathrm{M}_3)$ is a "w1" and, therefore, the state of all cells before performing $\mathrm{M}_2(\mathrm{M}_4)$ is 1, as required by the first march element of Case A.1 (Case A.2).

Condition lCFwd is satisfied as follows. For $x = 0$ (i.e., for $\mathrm{CFwd}_{0,0}$ and $\mathrm{CFwd}_{0,1}$), M_0 and M_1 of the test contain the first pair of march elements of Case A.1 and of Case B.1; and M_2 and M_3 contain the second pair of Case A.1 and of Case B.1. For $x = 1$ (i.e., for $\mathrm{CFwd}_{1,0}$ and $\mathrm{CFwd}_{1,1}$), M_1 and M_2 of the test contain the first pair of Case A.1 and of Case B.1; and M_3 and M_4 contain the second pair of Case A.1 and of Case B.1.

It is interesting to note that the last operation of M_4 (i.e., "w0") can be removed without having any impact on the fault coverage of the test. The operation is added to hold the symmetrical structure of the test which is suitable from the industrial point of view; it facilitate the implementation of the test.

2) Testing $\mathrm{LF2}_{\mathrm{av}}$ *Faults:* The set of $\mathrm{LF2}_{\mathrm{av}}$ faults consist of 16 faults; see Table VI. The detection of $\mathrm{LF}_{\mathrm{av}} = \mathrm{FP}_1 \rightarrow \mathrm{FP}_2$ requires the detection of one of the FPs in isolation. If one chooses to detect FP_1, then there will be 8 two-cell FPs to be detected, as Table VI shows; however, if one chooses to detect FP_2, then there will be 4 single-cell FPs to detect (which are $\mathrm{WDF}_1, \mathrm{WDF}_0, \mathrm{RDF}_1$, and RDF_0). Therefore, the detection of single-cell FP_2 will be considered. Note that the detection of single-cell FPs is easier than the detection of two-cell FPs, because they generally require shorter tests.

a) Test Conditions for $\mathrm{LF2}_{\mathrm{av}}$*:* From the above, it will be clear that the detection of $\mathrm{LF2}_{\mathrm{av}}$ can be done by detecting WDF and RDF in isolation. Although the same requirement as that for the detection of LF1s would be used; the detection condition for $\mathrm{LF2}_{\mathrm{av}}$ will be different because LF1s involve only a single cell. Hence, the detection of WDF and RDF in isolation for LF1s needs to consider only the v-cell, and detect the fault before another operation is performed to the same cell to prevent masking. However, $\mathrm{LF2}_{\mathrm{av}}$ involves two cells: the a-cell and the v-cell. An operation applied to the a-cell, before the v-cell is accessed, can cause masking like in the case of $\mathrm{CFds}_{xOy;0} \rightarrow \mathrm{WDF}_1$. In addition, accessing the v-cell with successive operations while the a-cell is in a certain state, can also cause masking as is the case for $\mathrm{CFtr}_{x;0} \rightarrow \mathrm{WDF}_1$.

The detection of all $\mathrm{LF2}_{\mathrm{av}}$ faults can be thus done by detecting $\mathrm{WDF}_0, \mathrm{WDF}_1, \mathrm{RDF}_0$ and RDF_1 in isolation. To detect $\mathrm{WDF}_0 = \langle 0w0/1/-\rangle$ involved in the $\mathrm{LF2}_{\mathrm{av}}$, a march test has to contain the following march element:

$$\Updownarrow (\ldots, O1, w0, w0, r0, \ldots)$$

where O can be a read or a write, and $O1$ guarantees that the content of the v-cell is 1 before the first "w0" operation is applied; such that the first write operation will be down transition and the second "w0" operation will be a nontransition "w0" operation, as required to sensitize WDF_0. The fault will be then detected by 'r0' operation which will return a wrong value 1 rather than the expected value 0. In order to prevent sensitizing $\mathrm{CFds}_{xOy;1}$ that can be linked to WDF_0, the march element can not be split (see Table VI). For example, if the march element is split into $\Updownarrow (\ldots, O1); \Updownarrow (w0, w0, r0, \ldots)$, then the $\mathrm{CFds}_{xOy;1} \rightarrow \mathrm{WDF}_0$ where "$y = 1$" may not be detected since the $\mathrm{CFds}_{xOy;1}$ may be sensitized by "$O1$" and masked by $\Updownarrow (w0, w0, r0, \ldots)$ due to WDF_0.

In the march element $\Updownarrow (\ldots, O1, w0, w0, r0, \ldots)$, required to detect WDF_0 in isolation, the "$O1$" can be:

1) A transition write operation (0w1): this can sensitize a $\mathrm{CFtr}_{x;1}$ which can be linked to WDF_0 (L_4 of Table VI). Masking will take place and the fault will be not detected.
2) A nontransition write operation (1w1): this can sensitize a $\mathrm{CFwd}_{x;1}$ which can be linked to WDF_0 (L_6 of Table VI). Masking will take place and the fault will be not detected.
3) A read operation (1r1): this can sensitize a $\mathrm{CFdr}_{x;1}$ which can be linked to WDF_0 (L_8 of Table VI). The fault will then not be detected due to the masking.

By allowing "$O1$" to be "$r1$," the $\mathrm{CFtr}_{x;1} \rightarrow \mathrm{WDF}_0$, the $\mathrm{CFwd}_{x;0} \rightarrow \mathrm{WDF}_0$ and the $\mathrm{CFds}_{yOx;1} \rightarrow \mathrm{WDF}_0$ will be detected by detecting WDF_0 in isolation and using the same march element. However, $\mathrm{CFdr}_{x;1} \rightarrow \mathrm{WDF}_0$ will be not detected since the "$r1$" can sensitize a $\mathrm{CFdr}_{x;1}$ that can be linked to WDF_0. Adding a second "$r1$" after "$O1 = r1$" and before the first "w0" in $\Updownarrow (\ldots, O1, w0, w0, r0, \ldots)$ will detect $\mathrm{CFdr}_{x;1} \rightarrow \mathrm{WDF}_0$ by detecting the $\mathrm{CFdr}_{x;1}$ in isolation. The test condition detecting all $\mathrm{FP}_1 \rightarrow \mathrm{WDF}_0$ will, therefore, become

$$\Updownarrow (\ldots, r1, r1, w0, w0, r0, \ldots).$$

If the state of the a-cell is "x" when the operations are applied to the v-cell, then the fault is detected by detecting $\mathrm{CFdr}_{x;1}$; the latter will be sensitized by the first "$r1$" operation and detected by the second one. If the sate of the a-cell was "$\bar{x}$," then the $\mathrm{CFdr}_{x;1}$ will not be sensitized since it required the a-cell to be in the state "x." In that case, the linked fault is detected by detecting WDF_0; the latter will be sensitized by the "$w0, w0$" and detected by "$r0$" operation.

Note that the same detection condition can also detect all $\mathrm{FP}_1 \rightarrow \mathrm{RDF}_0$. In fact only the march element $\Updownarrow (\ldots, w0, r0, \ldots)$ is required in order to detect this fault. Applying a "w0" before the read operation will overwrite any previous possible fault in the v-cell; and therefore it isolates the detection of RDF_0. There is no possible FP sensitized by that w0 operation that can be linked to RDF_0, therefore, no masking is possible.

$$\{ \underbrace{\Updownarrow (w0)}_{M_0} ; \underbrace{\Updownarrow (r0, r0, w1, w1, r1)}_{M_1} ; \underbrace{\Updownarrow (r1, r1, w0, w0, r0)}_{M_2} \}$$

Fig. 9. March LF2av.

In a similar way, one can construct the detection condition for $\mathrm{FP}_1 \rightarrow \mathrm{WDF}_1$ and $\mathrm{FP}_1 \rightarrow \mathrm{RDF}_1$. The results are summarized in the following.

Condition *LF2av*: Any $\mathrm{LF2}_{\mathrm{av}}$ is detectable by a march test which contains the march element of Cases A and of B.

- Case A: To detect $\mathrm{FP}_1 \rightarrow \mathrm{WDF}_0$ and $\mathrm{FP}_1 \rightarrow \mathrm{RDF}_0$

$$\Updownarrow (\ldots, r1, r1, w0, w0, r0, \ldots)$$

- Case B: To detect $\mathrm{FP}_1 \rightarrow \mathrm{WDF}_1$ and $\mathrm{FP}_1 \rightarrow \mathrm{RDF}_1$

$$\text{C.2. } \Updownarrow (\ldots, r0, r0, w1, w1, r1, \ldots).$$

b) Test for $\mathrm{LF2}_{\mathrm{av}}$*:* The above condition is compiled into a march test given in Fig. 9. It is referred to as *March LF2av* and has a test length of $11n$. March LF2av satisfies Condition $\mathrm{LF2}_{\mathrm{av}}$ as follows. M_2 contains the march element of Case A and M_1 contains the march element of Case B. Note that March LF2av is the same as March LF1 (see Fig. 7); therefore, they have the same fault coverage. However, in general any test detecting $\mathrm{LF2}_{\mathrm{av}}$'s also detects LF1s, but not vice-versa (see Condition LF1 and Condition LF2av).

3) Testing $\mathrm{LF2}_{\mathrm{va}}$ *Faults:* The set of $\mathrm{LF2}_{\mathrm{va}}$ faults consist of 18 faults (see Table VII). To detect $\mathrm{LF}_{\mathrm{va}} = \mathrm{FP}_1 \rightarrow \mathrm{FP}_2$, one of the FPs has to be detected in isolation. There are 6 single-cell FPs for FP_1 and 6 two-cell FPs for FP_2. Since the detection of single-cell FP is easier and requires shorter tests, the detection of the single-cell FP_1 will be considered.

a) Detection Conditions for $\mathrm{LF2}_{\mathrm{va}}$*:* Based on the above, one can conclude that the detection of all $\mathrm{LF2}_{\mathrm{va}}$ faults can be done by detecting TF, WDF, and DRDF faults in isolation. Since the $\mathrm{LF2}_{\mathrm{va}}$ faults involve two cells, one has to take into consideration the position of the two cells (a-cell and v-cell) in order to prevent the masking. A $\mathrm{LF2}_{\mathrm{va}}$ is sensitized by first accessing the v-cell (to sensitize TF, WDF, or DRDF) and thereafter the a-cell (to sensitize, e.g., CFrd). The $\mathrm{LF2}_{\mathrm{va}}$ can be detected by first accessing the v-cell to sensitize and detect the TF, the WDF and the DRDF in isolation, before proceeding to access the a-cell.

Condition *LF2va*: Any $\mathrm{LF2}_{\mathrm{va}}$ is detectable by a march test which contains one of two march elements of Case A, one of Case B, one of Case C, and one of Case D:

- Case A:
  (To detect $\mathrm{TF}_0, \mathrm{WDF}_0$, and DRDF_0 when $v < a$)
  A.1 $\Uparrow (\ldots, O1, w0, w0, r0, r0, \ldots)$
  A.2 $\Downarrow (\ldots, O1); \Uparrow (w0, w0, r0, r0, \ldots)$
- Case B:
  (To detect $\mathrm{TF}_0, \mathrm{WDF}_0$, and DRDF_0 when $v > a$)
  B.1 $\Downarrow (\ldots, O1, w0, w0, r0, r0, \ldots)$
  B.2 $\Uparrow (\ldots, O1); \Downarrow (w0, w0, r0, r0, \ldots)$
- Case C:
  (To detect $\mathrm{TF}_1, \mathrm{WDF}_1$, and DRDF_1 when $v < a$)
  C.1 $\Uparrow (\ldots, O0, w1, w1, r1, r1, \ldots)$
  C.2 $\Downarrow (\ldots, O0); \Uparrow (w1, w1, r1, r1, \ldots)$

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

Fig. 10. March LF2va.

- Case D:
  (To detect $\mathrm{TF}_1, \mathrm{WDF}_1$, and DRDF_1 when $v > a$)

D.1 $\Downarrow (\ldots, O0, w1, w1, r1, r1, \ldots)$

D.2 $\Uparrow (\ldots, O0); \Downarrow (w1, w1, r1, r1, \ldots)$.

The march element of Case A.1 will sensitize and detect $\mathrm{TF}_0 = \langle 1w0/1/-\rangle, \mathrm{WDF}_0 = \langle 0w0/1/-\rangle$, and $\mathrm{DRDF}_0 = \langle 0r0/1/0\rangle$ in isolation when the v-cell of two-cell FP involved in $\mathrm{LF2}_{\mathrm{va}}$ has a lower address than the a-cell $(v < a)$. The v-cell will thus be accessed first and tested before accessing the a-cell. The "$O1$" guarantees that the v-cell is set to 1. The first "w0" operation will sensitize TF_0; the second "w0" operation will sensitize WDF_0; the "r0" will detect both faults and sensitizes the DRDF_0, which will be detected by the followed "r0" operation. It is interesting to note that $\mathrm{TF}_0, \mathrm{WDF}_0$, and DRDF_0 have the same fault effect. Hence, no masking is possible. Case A.2 has the same capabilities as Case A.1, the only difference is that the operations of Case A.1 are divided over two march elements. Note that the first march elements of Case A.2 has a down addressing in order to guarantee that the v-cell has been accessed last and is initialized to 1, while in the second march element the v-cell will be accessed first; this will prevent possible masking.

Case B is required to detect $\mathrm{TF}_0, \mathrm{WDF}_0$, and DRDF_0 in isolation when $v > a$. Case C and D are required to detect $\mathrm{TF}_1, \mathrm{WDF}_1$, and DRDF_1 in isolation when $v < a$, respectively $v > a$.

b) Test for $\mathrm{LF2}_{\mathrm{va}}$*:* The test detecting all $\mathrm{LF2}_{\mathrm{va}}$ faults is shown in Fig. 10, and referred as *March LF2va*. It has a test length of $19n$, and satisfies Condition $\mathrm{LF2}_{\mathrm{va}}$ as follows. M_1 contains the march element of Case A.1; M_1 and M_2 contain the two march elements of Case D.2; M_3 contains the march element of Case C.1; while M_3 and M_4 contain the two march elements of Case B.2.

C. Testing Three-Cell LFs

Three-cell LFs describe linking two two-cell FPs with different a-cells and the same v-cell. The instances of this class of LFs are exactly the same as those for two-cell $\mathrm{LF2}_{\mathrm{aa}}$; see Table V. However, the compatibility condition does not apply for LF3s, since the two a-cells are different.

1) Detection Conditions for LF3s: Since the FPs forming LF3s (i.e., CFds, CFtr, CFwd, CFrd, and CFdr) are the same as those forming $\mathrm{LF2}_{\mathrm{aa}}$ faults, the same test conditions apply. Therefore, LF3s can be tested with the same test as that introduced for $\mathrm{LF2}_{\mathrm{aa}}$ faults; i.e., March LF2aa given in Fig. 8 detects all $\mathrm{LF2}_{\mathrm{aa}}$ as well as all LF3 faults. Hence, Condition lCFwd (and Condition lCFrd) designed for $\mathrm{LF2}_{\mathrm{aa}}$ (see Section VI-B) also apply for LF3 faults; however, Condition lCFds developed for $\mathrm{LF2}_{\mathrm{aa}}$ is too strong for LF3s. The two FPs involved in $\mathrm{LF2}_{\mathrm{aa}}$ have the same a-cell as well as the same v-cell; therefore performing a sequence of operations to the same a-cell can sensitize (e.g., due to FP_1) and mask (e.g., due to FP_2) the fault; this is not the case for LF3 since two different a-cells are involved. Therefore Condition CFds presented in Section VI-A can be used to detect all LF3s rather than Condition lCFds, which is too strong for LF3s. Below Condition CFds is re-given and called Condition lCFds for LF3s.

<u>*Condition*</u> <u>*lCFds*</u> (for LF3s): A march test which contains the following four march elements detects all CFds (involved in LF3: CFds → CFds) in isolation:

- Case A: to detect $\mathrm{CFds}_{xOy;0} = \langle xOy; 0/1/-\rangle$:

A.1 If $v > a$:$\Uparrow (r0, w1, w1, r1, w0, w0, \ldots)$;

A.2 If $v < a$:$\Downarrow (r0, w1, w1, r1, w0, w0, \ldots)$;

- Case B: to detect $\mathrm{CFds}_{xOy;1} = \langle xOy; 1/0/-\rangle$:

B.1 If $v > a$:$\Uparrow (r1, w0, w0, r0, w1, w1, \ldots)$;

B.2 If $v < a$:$\Downarrow (r1, w0, w0, r0, w1, w1, \ldots)$;

For the explanation of the above detection condition see Section VI-B.

2) Test for LF3s: Based on the above condition and Condition lCFwd (see Section VI-B), a test detecting all LF3s can be derived. Note: Condition lCFrd is a subset of Condition lCFwd; hence any test detecting the FPs linked to CFwd also detects the FPs linked to CFrd. The result, referred to as *March LF3*, is given in Fig. 11 and has a test length of $38n$. It can be easily seen that the test satisfies Condition lCFds for LF3s by $\mathrm{M}_1, \mathrm{M}_2, \mathrm{M}_4$, and M_5. In addition, Condition lCFwd (see Section 6.2.1) is satisfied as follows. For $x = 0, \mathrm{M}_0$ and M_1 contain the first pair of Case A.1 and of Case B.1; and M_1 and M_2 contain the second pair of Case A.1 and of Case B.1. For $x = 1, \mathrm{M}_3$, and M_4 contain the first pair of Case A.1 and of Case B.1; and M_4 and M_5 contain the second pair of Case A.1 and of Case B.1.

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

Fig. 11. March LF3.

TABLE VIII
SUMMARY OF LINKED FAULTS TESTS

Tests	T.L.	LF1	LF2aa	LF2av	LF2va	LF3
March LF1	$11n$	+	-	+	-	-
March LF2aa	$41n$	+	+	+	+	+
March LF2av	$11n$	+	-	+	-	-
March LF2va	$19n$	+	-	-	+	-
March LF3	$38n$	+	-	+	+	+
March SL	$41n$	+	+	+	+	+

D. Test for All LFs

Table VIII summarizes all tests introduced in this section. It shows their required test length (T.L.), including the initialization, together with their fault coverage. In the table, "+" indicates that the test detects the corresponding LF, and "−" indicates that the test does not cover, or only partially covers, the corresponding LFs. For example, March LF1 detects all LF1

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

Fig. 12. March SL.

faults and $\mathrm{LF2_{av}}$ faults; however, it does not detect (all) LF3 faults. The evaluation of each test is done by verifying whether the test satisfy the detection condition(s) of the corresponding linked fault (sub)class.

The table shows clearly that the test with highest fault coverage is March $\mathrm{LF2_{aa}}$. The test is re-given in Fig. 12, and referred to as **March SL**: a test for *all static LFs*, based on two simple faults. March SL also detects all simple faults of Tables I and II [9]. March SL covers all static, simple and LFs in single-port RAMs; see the scope shown in Fig. 1.

- All LF1s are detected. This is because Condition LF1 of Section VI-A is satisfied by $\mathrm{M_1}$ (as well as by $\mathrm{M_2, M_3}$, and $\mathrm{M_4}$).
- All $\mathrm{LF2_{aa}}$s are detected since March LF is the same as March LF2aa.
- All $\mathrm{LF2_{av}}$s are detected. Condition $\mathrm{LF2_{av}}$ is satisfied by $\mathrm{M_1}$ (as well as by $\mathrm{M_2, M_3}$, and $\mathrm{M_4}$).
- All $\mathrm{LF2_{va}}$s are detected since Condition LF2va is satisfied. $\mathrm{M_1}$ contains the march element of Case C.1; $\mathrm{M_2}$ contains the march element of Case A.1; $\mathrm{M_3}$ contains the march element of Case D.1; and $\mathrm{M_4}$ contains the march element of Case B.1.
- All LF3s are detected since Condition LF3 is satisfied. $\mathrm{M_1}$ contains the march element of Case A.1; $\mathrm{M_2}$ contains the march element of Case B.1; $\mathrm{M_3}$ contains the march element of Case A.2; and $\mathrm{M_4}$ contains the march element of Case B.2.

VII. Evaluation of March SL

This section first gives a quantitative analysis of the traditional tests regarding their capability in detecting LFs discussed in this paper. Thereafter, the test results of DPM screening done at Intel for advanced high speed caches will be presented; they will validate the high fault coverage of March SL against any existing memory test.

A. Analytical Evaluation

Table IX summarizes the fault coverage of the most known memory tests for LFs. The test length of each test is given; n denotes the size of the memory, R denotes the number of rows and C denotes the number of columns. Note that there are two classes of test algorithms:

1) $O(n)$ class: These are linear tests; they consist of test 1 through test 6 [5], [7], [12], [22].
2) $O(n^2)$ class: These consist of two tests; the known Galpat and Walking $1/0$.

At the end of the table, March SL is added.

In the table, e.g., March C- detects 10 of 12 LF1 faults of Table IV, 10 of 24 $\mathrm{LF2_{aa}}$ faults of Table V, 11 of 16 $\mathrm{LF2_{av}}$ faults of Table VI, 9 of 18 $\mathrm{LF2_{va}}$ faults of Table VII, and 10 of 24 LF3 faults. The total LFs detected by March C- is then 50 from 94. The table clearly shows that a test with a higher test length does not necessarily have a higher fault coverage for LFs. In addition, it shows that none of the tests can cover all considered LFs, and that the test scoring the best is March SR [7] with $\mathrm{FC} = 72/94$, followed with PMOVI with a FC of $63/94$. Note that even with all traditional tests of Table IX, a 100% FC of LFs can not be reached. The newly introduced March SL, with a test length of $41n$, has a 100% linked fault coverage.

B. Industrial Evaluation

The tests shown in Table IX have been implemented and applied to Intel advanced SRAM's chips. The SRAM memory considered in this experiment has a size of 512 KBytes, with a word size of $B = 32$ bits, and is tested at 200 MHz. Each cell is a standard 6-transistor CMOS SRAM cell.

1) Used Stress Combinations: Each test has been applied using different stress combinations. A stress combination specifies the way the test is performed, and therefore they influence the sequence and/or the type of the memory operations. The used stresses are the address orders and the data-backgrounds.

The used addressing stresses consist of two types of addressing:

1) **Fast X (fx)**: Fast X addressing is simply incrementing or decrementing the address in such a way that each step goes to the next row. The sequence should proceed in the physical order of the word lines.
2) **Fast Y (fy)**: Fast Y addressing is simply incrementing or decrementing the address in such a way that each step goes to the next column. The sequence should proceed in the adjacent physical order of the bit lines.

A *data-background (DB)* is the pattern of ones and zeros as seen in an array of memory cells. The most common types of data-background are four: solid, checkerboard, column strip, and row strip. Fig. 13 illustrates the four DBs using a simple 4×4 array. Each DBs is shown in with the base and the complement values.

2) Test Results: Table X lists the 40 total number of tests applied. A test consists of a base test (BT) (e.g., March C-) applied using a particular stress combination (SC). The total number of tests is 40; that is the sum of SCs for all 11 BTs. Note that Galpat and Walking $1/0$ are each implemented two times in their $O(n\sqrt{n})$ versions, which restrict the read operation actions only to the same column (e.g., GalColumn) or to the same row as the *base cell*. The column "#SC" gives the number of SCs each BTs is used with. In the table, the solid, the checkerboard, column strip and row strip data background are denoted as "s", "c", "cs", and "rs" respectively; while the addressing are denoted as "fx" and "fy." A "+" in the table indicates that the corresponding SC is implemented, and a "−" denoted that it is not. For example, GalColumn is implemented using "fx-s."

All SCs have been implemented at high-voltage power supply. From a huge volume of SRAM chips tested at wafer level and low temperature, 1577 chips failed; 1343 chips do

TABLE IX
ANALYTICAL COMPARISON OF THE MEMORY TESTS.

#	Tests	Test Length	LF1	$LF2_{aa}$	$LF2_{av}$	$LF2_{va}$	LF3	Total FC
1	Scan	4n	8/12	4/24	8/16	3/18	4/24	27/94
2	Mats+	5n	8/12	0/24	8/16	1/18	0/24	17/94
3	Mats++	6n	10/12	0/24	8/16	4/18	0/24	22/94
4	March C-	10n	10/12	10/24	11/16	9/18	10/24	50/94
5	PMOVI	13n	10/12	15/24	11/16	12/18	15/24	63/94
6	March SR	14n	10/12	16/24	13/16	15/18	18/24	72/94
7	Walking 1/0	8n+2nRC	8/12	12/24	9/16	10/18	12/24	51/94
8	Galpat	6n+4nRC	10/12	13/24	9/16	14/18	13/24	59/94
9	**March SL**	4in	12/12	24/24	16/16	18/18	24/24	**94/94**

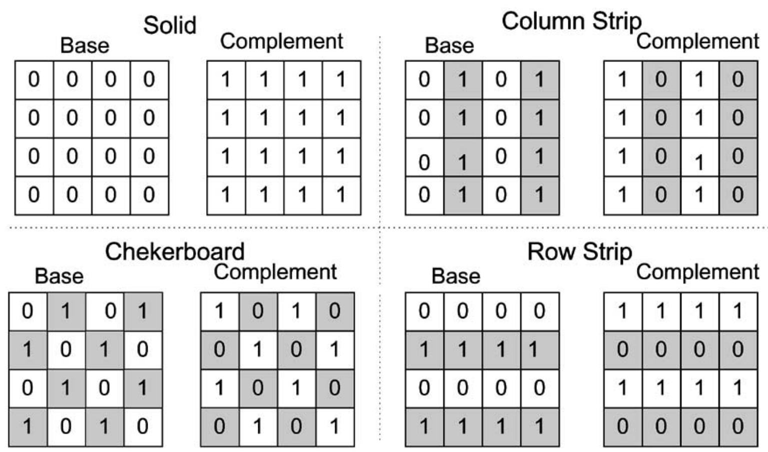


Fig. 13. Common data-backgrounds.

TABLE X
LIST OF USED STRESS COMBINATIONS

#	Base Test (BT)	# SCs	Stress combination fx s	fx c	fx cs	fx rs	fy s	fy c	fy cs	fy rs
1	GalColumn	1	+	-	-	-	-	-	-	-
2	GalRow	1	-	-	-	-	+	-	-	-
3	March C-	4	+	-	+	-	+	-	+	-
4	Mats+	2	+	-	-	-	+	-	-	-
5	Mats++	2	+	-	-	-	+	-	-	-
6	PMOVI	8	+	+	+	+	+	+	+	+
7	Scan	4	-	+	+	+	-	+	-	-
8	March SL	8	+	+	+	+	+	+	+	+
9	March SR	8	+	+	+	+	+	+	+	+
10	WalkColumn	1	+	-	-	-	-	-	-	-
11	WalkRow	1	-	-	-	-	+	-	-	-

fail all SCs (i.e., 40) while 201 chips do not. From now one, we will only concentrate on SRAM chips that do not fail all SCs since they are the most important.

Since the data base of the test results is very large, it has to be simplified for analysis purposes. Therefore, we will first consider the fault coverage of each BT. The fault coverage of an BT is the *union* of the fault coverages of its corresponding SCs. A die belongs to the union if at least one SC of that BT found the die to be faulty. For example, Mats+ is implemented using fx-s (i.e., Fast X, solid data-background) and fy-s; the fault is considered detected if at least one of the two implementations of Mats+ detects the fault.

Table XI shows the intersections and the union of the 11 BTs. A die belongs to the intersection of two BTs if both BTs found the die to be faulty, and belongs to the union of two BTs, if at least one of the two BTs found the die to be faulty. The first column in each table gives the BT number; the second the column name of the BT. The column "FC" lists the fault coverage of the corresponding BT; the column "UFs" gives number of unique faults (UFs) each BT detects. Unique faults are faults that are only detected once with a single test; e.g., GalRow detects nine unique faults that are not detected with any other test. The intersection and the union of each pair of BTs is shown in the rest of the table. The numbers on the diagonal give the

TABLE XI
INTERSECTION AND UNIONS OF BTs (TOTAL DETECTED FAULTY CHIPS IS 201)

#	Test	FC	UFs	1	2	3	4	5	6	7	8	9	10	11
1	GalColumn	164	0	**164**	152	*164*	159	163	162	155	163	153	*160*	151
2	GalRow	176	9	188	**176**	167	162	165	167	154	166	154	148	166
3	March C-	183	0	183	192	**183**	174	*177*	179	165	182	166	*160*	*168*
4	MATS+	175	0	180	189	184	**175**	173	173	162	*175*	160	156	162
5	MATS++	177	0	178	188	183	179	**177**	175	164	176	162	*160*	164
6	PMOVI	181	1	183	190	185	183	183	**181**	164	179	164	158	*168*
7	Scan	168	1	177	190	186	181	181	185	**168**	165	160	155	155
8	March SL	185	3	186	*195*	186	185	186	187	188	**185**	167	159	167
9	March SR	170	1	181	192	187	185	185	187	178	188	**170**	152	154
10	WalkColumn	160	0	164	188	183	179	177	183	173	186	178	**160**	147
11	WalkRow	168	0	181	178	183	181	181	181	181	186	184	181	**168**

TABLE XII
BTs DETECTING UNIQUE FAULTS

BT	FC	# UFs
GalRow	176	9
PMOVI	181	1
Scan	168	1
March SL	185	3
March SR	170	1

fault coverage (FC) of the BTs, which are also listed in the column "FC;" e.g., Mats+ has FC = 175. The part above the main diagonal shows the intersection for each BT pair, while the part under the diagonal lists the union of each BT pair; for example, the intersection of March C- and PMOVI is 179 and their union is 185. Remember that the FC presented in Table XI for each BT presents the union of the fault coverages of its corresponding SCs. Based on the data of the table, one can conclude the following:

- Total number of faulty chips detected: 201.
- The best BTs, in terms of FC, is March SL. However, this is a strong conclusion since not all BTs used the same SCs.
- There are 15 unique faults, detected by 5 tests listed in Table XII with their FC and the number of unique faults (# UFs) each detects.

  It has been shown that LFs can not be covered with any of the traditional tests, therefore March SL has been designed to target LFs. The detected UFs by March SL are (probably) LFs. Other BTs detecting UFs need more detailed analysis. These BTs detect faults that can not be explained using the well-known fault models. This means that additional fault models and/or fault classes exist. By analyzing each failed test for UFs, new fault models will be introduced; this methodology has also been used in our previous work [10] to introduce March RAW; which is in use in the production test at Intel and has been shown to be very effective. Failure analysis can also be used for better understanding of the underlying defect causing such SF behavior. A new test(s) with a shorter test length, targeting the detected UFs and replacing the nonlinear empirical tests, can be constructed and used for further test purposes, instead of expensive ones.
- Some BTs detect supersets of faults of other BTs. For example: GalColumn detects a superset of WalkColumn.

TABLE XIII
LIST OF BTs DETECTING SUPERSETS OF OTHER ONES

BT	Covered BTs
GalColumn	WalkColumn
March C-	Mats++, WalkColumn, WalkRow
MATS++	WalkColumn
PMOVI	WalkRow
March SL	Mats+

TABLE XIV
SCs INTERSECTIONS AND UNIONS FOR MARCH SL

SC	fx-c	fx-cs	fx-rs	fx-s	fy-c	fy-cs	fy-rs	fy-s
(fx,c)	**175**	173	172	172	173	174	173	172
(fx,cs)	178	**176**	**173**	172	173	175	173	172
(fx,rs)	176	176	**173**	172	171	172	172	172
(fx,s)	177	178	175	**174**	171	173	171	172
(fy,c)	181	182	181	182	**179**	**179**	178	177
(fy,cs)	183	183	183	183	182	**182**	178	178
(fy,rs)	182	183	181	183	181	**184**	**180**	178
(fy,s)	182	183	180	181	181	183	181	**179**

  Table XIII summarizes the BTs detecting supersets of faults of other ones. The initial 11 BTs of Table X can therefore be reduced to seven BTs while achieving the same fault coverage; that means that the BTs covered by other ones can be removed.
- The best union pair of BTs, in terms of FC, is the union of March SL and GalRow with FC = 195.
- An analysis (not shown here) reveals that the optimal BT set required for 100% FC consists only of five BTs shown in Table XII.

3) Effect of Stresses on March SL: Table XIV shows the effect of the stresses on March SL. The first column lists the use SC; e.g., (fx, c) denotes Fast X addressing and Checkerboard data-background. The rest of the table prints the effect of the impact of the stresses on the FC. The main diagonal, which is printed in bold, gives the FC of each SC; the part above the main diagonal lists the intersection of each pair of SCs, while the part below the diagonal lists the unionof each pair of SCs. Remember that the maximal FC that have been achieved with all SCs for March SL is FC = 185. Based on the data of Table XIV, the following can be concluded:

- The (fy, cs) is the most effective SC detecting 182/185.
- The union of (fy, cs) and (fy, rs) is the most effective SC pair union detecting 184/185 faults.

- The (fx, cs) covers the superset of (fx, rs); and the (fy, cs) covers that of (fy, c).
- An analysis (not shown here) reveals that a least three SCs are required in order to detect $185/185$ faults; these can be one of the following three triple of SCs: $[(\mathrm{fx},\mathrm{cs}),(\mathrm{fx},\mathrm{s}),(\mathrm{fy},\mathrm{rs})]$, $[(\mathrm{fx},\mathrm{cs}),(\mathrm{fx},\mathrm{s}),(\mathrm{fy},\mathrm{s})]$, or $[(\mathrm{fx},\mathrm{s}),(\mathrm{fy},\mathrm{cs}),(\mathrm{fy},\mathrm{rs})]$,

VIII. Conclusion

In this paper, a complete analysis for LFs in RAMs has been presented based on the FP concept. After having giving the whole space of simple faults, a precise definition of LFs has been introduced and used to establish their whole fault space, by assuming that a LF is a combination of two simple faults. This space has been divided, based on the number of cells required in the LF, into five (sub)classes. Thereafter, a methodology to design tests for each (sub)class has been introduced, by first developing detection conditions, and then compiling them into tests. The result is a set of five linear march tests. An analytical evaluation of five tests reveals that one of the tests has also the capability to detect all other linked fault (sub)classes. This test, which is named as *March SL*, can be thus used to cover all single-port, static, LFs in RAMs.

Finally, March SL has been evaluated analytically as well as industrially. The preliminary test results show that it has a high fault coverage as compared with the other traditional tests. In addition, they show that March SL detects some unique faults (probably LFs) that can not be detected with any other tests, which indicates the superiority of March SL. Moreover, the test results show that there are some detected unique faults which cannot be explained with the traditional fault models. Such fault remain still to be explained; this will allow eliminating the nonlinear algorithms that detect such faults with linear optimized tests for specific faults.

References

[1] M. S. Abadir and J. K. Reghbati, "Functional testing of semiconductor random access memories," *ACM Comput. Surveys*, vol. 15, no. 3, pp. 175–198, 1983.

[2] 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, pp. 27–32.

[3] Z. Al-Ars and A. J. van de Goor, "Impact of memory cell array bridges on the faulty behavior in embedded drams," in *Proc. ofAsian Test Symp.*, 2000, pp. 282–289.

[4] ——, "Static and dynamic behavior of memory cell array opens and shorts in embedded DRAMs," in *Proc. Design Automation Test Eur.*, 2001, pp. 496–503.

[5] M. A. Breuder and A. D. Friedman, *Diagnosis and Reliable Design of Digital Systems*. Woodland Hills, CA, USA: Computer Science, 1976.

[6] R. Dekker, F. Beenaker, and L. Thijssen, "A realistic fault model and test algorithm for static random access memories," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 567–572, June 1990.

[7] S. Hamdioui and A. J. van de Goor, "Experimental analysis of spot defects in SRAMs: Realistic fault models and tests," in *Proc. 9th Asian Test Symp.*, 2000, pp. 131–138.

[8] ——, "Efficient tests for realistic faults in dual-port SRAMs," *IEEE Trans. Comput.*, vol. 51, pp. 460–473, May 2002.

[9] S. Hamdioui, "RAM-LFs-Testing Research Project," Delft Univ. Technol., Comput. Eng. Lab. Tech. Rep., Delft, The Netherlands, 2002.

[10] S. Hamdioui, Z. Al-Ars, and A. J. van de Goor, "Testing static and dynamic faults in random access memories," in *Proc. VLSI Test Symp.*, 2002, pp. 395–400.

[11] V. K. Kim and T. Chen, "On comparing functional fault coverage and defect coverage for memory testing," *IEEE Trans. Computer-Aided Design*, vol. 18, pp. 1676–1683, Nov. 1999.

[12] M. Marinescu, "Simple and efficient algorithms for functional RAM testing," in *Proc. IEEE Int. Test Conf.*, 1982, pp. 236–239.

[13] M. Nicolaidis, A. C. Alves, and H. Bederr, "Testing complex coupling faults in multiport memories," *IEEE Trans. on VLSI Systems*, vol. 3, no. 1, pp. 59–71, Mar. 1995.

[14] R. Nair, "Efficient test algorithms for testing semiconductor random access memories," *IEEE Trans. Comput.*, vol. C-28, pp. 572–567, Mar. 1978.

[15] C. A. Papachristou and N. B. Saghal, "An improved method for detecting functional faults in random access memories," *IEEE Trans. Comput.*, vol. C34, pp. 110–116, Mar. 1985.

[16] *The National Roadmap for Semiconductors*, 2000. Semiconductor Industry Association.

[17] D. S. Suk and S. M. Reddy, "A march test for functional faults in semiconductor random access memories," *IEEE Trans. Comput.*, vol. C30, pp. 982–985, Dec. 1981.

[18] A. J. van de Goor and Z. Al-Ars, "Functional fault models: A formal notation and taxonomy," in *Proc. IEEE VLSI Test Symp.*, 2000, pp. 281–289.

[19] A. J. van de Goor *et al.*, "March LA: A test for linked memory faults," in *Proc. Eur. Design Test Conf.*, 1999, p. 627.

[20] A. J. van de Goor and A. Paalvast, "Industrial evaluation of DRAM SIMM tests," in *Proc. IEEE Int. Test Conf.*, 2000, pp. 426–435.

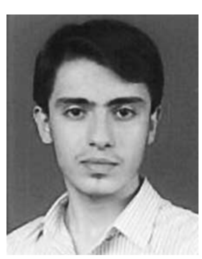
[21] A. J. van de Goor *et al.*, "March LR: A test for realistic linked faults," in *Proc. IEEE VLSI Test Symp.*, 1996, pp. 272–280.

[22] ——, *Testing Semiconductor Memories, Theory, and Practice*. Gouda, The Netherlands: ComTex, 1998.

Said Hamdioui (S'99–M'01) received the M.S.E.E. and Ph.D. degrees (both with honors) from the Delft University of Technology, Delft, The Netherlands, in 1997 and 2001, respectively.

He is currently with the Delft University of Technology. He interned with Intel Corporation for over two years, while pursuing and after receiving the Ph.D. degree, and was responsible for developing new low-cost and efficient test algorithms for advanced Intel singe-port and multiport cache designs. He has published about 25 papers in the area of testing. His research interests concern fault modeling, failure analysis, fault simulation, memory testing, test design, DFT, BIST, BISR, etc.

Dr. Hamdioui was the recipient of the European Design Automation Association (EDAA) Award for 2003.



Zaid Al-Ars (M'03) received the M.S. degree in electrical engineering with honors from the Delft University of Technology, Delft, the Netherlands, in 2000. He is working toward the Ph.D. degree in electrical engineering at the same university in cooperation with Infineon Technologies, Munich, Germany, where he is currently based.

He has published numerous papers in the field of electrical defect simulation, fault modeling, and test generation in memory devices. His research project involves systematic fault analysis, and test generation and optimization for commodity as well as embedded DRAM products.

Ad J. van de Goor (M'03) received the M.S.E.E. degree from the Delft University of Technology, Delft, the Netherlands, in 1965 and the M.S.E.E. and Ph.D. degrees from Carnegie-Mellon University, Pittsburgh, PA, in 1970.

He worked with Digital Equipment Corporation, Maynard, MA, as the chief architect of the PDP-11/45 computer. He also worked for IBM in the Netherlands and in the United States, being responsible for the architecture of embedded systems. He was a Professor of Computer Engineering at the Delft University of Technology and is currently retired. He has written two books and over 150 papers in the areas of computer architecture and testing. He is on the Editorial Board of the *Journal of Electronic Testing: Theory and Applications*. His main research interests are in testing memories and logic.

Mike Rodgers (S'84–M'86) received the B.S.E.E. degree from the University of Illinois and conducted additional studies at the University of Chicago.

He has been with Intel for 16 years. He has held various positions in Microprocessor Q&R and Test Technology, including running Q&R and FA groups in Santa Clara, CA, Japan, and the Chandler, AZ Test Factory. He has worked with DF∗ teams on all IA CPUs going since the 286, was primary driver for Intel's Cache DFT and test methods and now focuses on DFT and methods for manufacturing tests. He currently runs Test Planning and Integration in Design Technology in Santa Clara, CA and is in charge of Technology Planning for Test between technology, product, and factory teams.

Dr. Rogers co-chairs the Test TWG of the ITRS Roadmap. He is a member of the TTTC.