# A Fault Primitive Based Analysis of Linked Faults in RAMs

Zaid Al-Ars Said Hamdioui Ad J. van de Goor
Delft University of Technology, Faculty of Information Technology and Systems
Computer Engineering Laboratory, Mekelweg 4, 2628 CD Delft, The Netherlands
E-mail: z.e.al-ars@its.tudelft.nl

**Abstract:** *Linked faults are very important for memory testing because they reduce the fault coverage of the tests. Their analysis has proven to be a source for new memory tests, characterized by an increased fault coverage for a given test time. This paper presents an analysis of linked faults, based on the concept of fault primitives, such that the whole space of linked faults is investigated, accounted for and validated. The paper also introduces a systematic way to develop tests for such faults.*

**Keywords:** *Memory testing, fault primitives, functional fault models, linked faults, march tests.*

## 1 Introduction

Depending on the way memory faults manifest themselves, they can be divided into *simple faults* and *linked faults*.

**Simple faults**: These are faults that cannot influence the behavior of each other. This means that the behavior of a simple fault cannot change the behavior of another one, and therefore *masking* cannot occur.

**Linked faults**: These are faults that do influence the behavior of each other. This means that the behavior of a certain fault can change the behavior of another one such that *masking* can occur [8]. Note that linked faults consist of two or more simple faults. Due to masking, testing for linked faults is more complicated than testing for simple faults.

Theoretically, linked faults may take place in any memory containing multiple interacting faults, but since it is very complex to analyze linked faults, many memory tests are designed under the assumption that linked faults do not take place.

Practically, the importance of linked faults has been validated experimentally by [10], which describes the results of testing 800 DRAM chips with different march tests. In this study, a march test called March LA, designed specifically for detecting linked faults, had a higher fault coverage than other march tests. However, March LA was designed based only on the fault models which were known at that time. Experimental work, based on the injection of resistive defects in the electrical design of SRAMs [2, 4, 5, 6], and of DRAMs [3], has shown the existence of several new fault models.

This paper gives a precise definition of linked faults, based on the concept of fault primitives. In addition, it establishes the space of all possible linked faults and introduces a systematic way to design tests for such faults. The paper also gives a real example to show that linked faults take place practically.

The paper is organized as follows. Section 2 discusses *simple* faults which will be used for modeling *linked* faults. Section 3 introduces the concept of linked faults, while Section 4 establishes the space of linked faults and validates it. Section 5 develops test design methodology, and Section 6 ends with the conclusions.

## 2 Simple faults

This section starts with a description of the concept of fault primitives, which is then used to define the two most important types of simple faults: single-cell faults and two-cell faults (i.e., two-coupling faults) [2, 3, 4, 5, 6].

### 2.1 Concept of fault primitives

In order to specify a certain fault, one has to specify the *sensitizing operation sequence* ($S$), the corresponding *faulty behavior* ($F$) which is the observed memory behavior that deviates from the expected one, and the value read ($R$) from the faulty cell. The combination of $S$, $F$ and $R$ for a given memory fail is called a *fault primitive (FP)* [9], and is denoted as $<S/F/R>$.

The concept of FPs allows for establishing a complete framework of all memory faults, since for all allowed operation sequences, one can derive all possible types of faulty behavior. In addition, it makes it possible to give a precise definition of a *functional fault model (FFM)* [9]: *a functional fault model is a non-empty set of fault primitives.*

### 2.2 Single-cell simple faults

We start by a description of the FP notation to be used here, followed by a listing of the possible single-cell FPs.

$<S/F/R>$ (or $<S/F/R>_v$): denotes an FP involving a single-cell. The fault appears in the same cell ($v$ or *victim cell*) where the sensitizing operation is performed.

$S$ describes the value/operation *sensitizing* the fault; $S \in \{0, 1, 0w0, 1w1, 0w1, 1w0, 0r0, 1r1\}$, where 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* victim 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 the output data is not applicable, as it is for example the case if $S = 1w0$, where no data is expected to appear at the memory output, and therefore $R$ is replaced by a '$-$'.

Given all possible values of $S$, $F$, and $R$ for single-cell simple faults, it is possible to list all detectable FPs using the notation $<S/F/R>$. There are 12 possible FPs [9], compiled into a set of six functional fault models (FFMs) as listed in Table 1.

1. *State Fault (SF)*: A cell is said to have a *state fault* 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 and, therefore, it only depends on the initial stored value in the cell. The SF consists of two FPs: $<0/1/->$ and $<1/0/->$.
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)*.

**Table 1.** List of single-cell simple FFMs

| # | FFM | FP | FP description |
|---|---|---|---|
| 1 | SF | $SF_0$ | $<0/1/->$ |
|  |  | $SF_1$ | $<1/0/->$ |
| 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 | $IRF_0$ | $<0r0/0/1>$ |
|  |  | $IRF_1$ | $<1r1/1/0>$ |

## 2.3 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. Such FPs can be presented as $<S/F/R>$ = $<S_a; S_v/F/R>_{a,v}$, 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. Note that in the notation $<S_a; S_v/F/R>_{a,v}$, 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. $S_a, S_v \in \{0, 1, 0w0, 1w1, 0w1, 1w0, r0, r1\}$. As an example, $<S_a; S_v/F/R>_{a,v}$ = $<0w1; 0/1/->_{a,v}$ 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 $F = 1$), and since the output data is not applicable, $R$ is replaced with '$-$'. There are 36 possible two-cell FPs [9], compiled into seven FFMs listed in Table 2.

1. *State coupling fault (CFst)*: 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. For example, $<0; 0/1/->$: the v-cell will be forced to 1 if the a-cell is in the state 0.
2. *Disturb coupling fault (CFds)*: The CFds consists of 12 FPs, where an operation (read, transition or non-transition write) performed on the a-cell causes the v-cell to flip. Examples of CFds include $CFds_{xwy;0}$ = $<xwy; 0/1/->$ and $CFds_{xrx;1}$ = $<xrx; 1/0/>$ where $x, y \in \{0, 1\}$, $w$ 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)*.

**Table 2.** List of two-cell simple FFMs ($x, y \in \{0, 1\}$)

| # | FFM | FP | FP description |
|---|---|---|---|
| 1 | CFst | $CFst_{x;0}$ | $<0; 0/1/->$, $<1; 0/1/->$ |
|  |  | $CFst_{x;1}$ | $<0; 1/0/->$, $<1; 1/0/->$ |
| 2 | CFds | $CFds_{xwy;0}$ | $<xwy; 0/1/->$ |
|  |  | $CFds_{xwy;1}$ | $<xwy; 1/0/->$ |
|  |  | $CFds_{xrx;0}$ | $<0r0; 0/1/->$, $<1r1; 0/1/->$ |
|  |  | $CFds_{xrx;1}$ | $<0r0; 1/0/->$, $<1r1; 1/0/->$ |
| 3 | CFtr | $CFtr_{x;0}$ | $<0; 1w0/1/->$, $<1; 1w0/1/->$ |
|  |  | $CFtr_{x;1}$ | $<0; 0w1/0/->$, $<1; 0w1/0/->$ |
| 4 | CFwd | $CFwd_{x;0}$ | $<0; 0w0/1/->$, $<1; 0w0/1/->$ |
|  |  | $CFwd_{x;1}$ | $<0; 1w1/0/->$, $<1; 1w1/0/->$ |
| 5 | CFrd | $CFrd_{x;0}$ | $<0; 0r0/1/1>$, $<1; 0r0/1/1>$ |
|  |  | $CFrd_{x;1}$ | $<0; 1r1/0/0>$, $<1; 1r1/0/0>$ |
| 6 | CFdr | $CFdr_{x;0}$ | $<0; 0r0/1/0>$, $<1; 0r0/1/0>$ |
|  |  | $CFdr_{x;1}$ | $<0; 1r1/0/1>$, $<1; 1r1/0/1>$ |
| 7 | CFir | $CFir_{x;0}$ | $<0; 0r0/0/1>$, $<1; 0r0/0/1>$ |
|  |  | $CFir_{x;1}$ | $<0; 1r1/1/0>$, $<1; 1r1/1/0>$ |

# 3 Concept of linked faults

Let LF= $FP_1 \rightarrow FP_2$ denote the *Linked fault* LF, which consists of $FP_1$ *linked to* $FP_2$. The sensitizing operation sequence ($S_1$) of $FP_1$ is applied first, which sensitizes a fault in the v-cell. Next $S_2$ of $FP_2$ is applied, which also sensitizes a fault in the *same v-cell*, but with a fault effect *oppo-*

*site* to that of S$_1$. The net result is that the fault effect of FP$_2$ *masks* the fault effect of FP$_1$. In order to clarify the idea of linked faults (LFs), this section starts with two examples, followed by the definition of LFs, which will be used to derive their total space.

### 3.1 Understanding linked faults

**Example 1:** Consider LF$_1$ = FP$_1$ $\rightarrow$ FP$_2$ shown in Figure 1, where FP$_1$ = $<0; 0w1/0/->$ and FP$_2$ = $<0w1; 0/1/->$; note that FP$_1$ represents a FP of CFtr and FP$_2$ a FP of CFds (see Table 2). Assume that the addresses of the a-cells C$_{a1}$ and C$_{a2}$ and the v-cell C$_v$ have the following relationship: $v < a2 < a1$. Then, MATS+ test [1]: $\{\Updownarrow(w0);\ \Uparrow(r0, w1);\ \Downarrow(r1, w0)\}$ 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 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 FP$_1$ $\rightarrow$ FP$_2$ does not imply that FP$_2$ $\rightarrow$ FP$_1$ (i.e., the *linked to* relationship is *not commutative*).

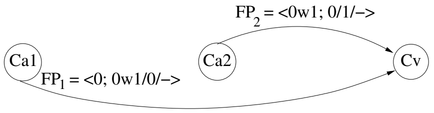

**Figure 1.** LF$_1$ = FP$_1$ $\rightarrow$ FP$_2$

**Example 2:** The reader can easily verify that the LF$_2$ shown in Figure 2 (where $a1 < a2 < v$) cannot be detected with the well-known March C- test [7]: $\{\Updownarrow(w0);\ \Uparrow(r0, w1);\ \Uparrow(r1, w0);\ \Downarrow(r0, w1);\ \Downarrow(r1, w0);\ \Updownarrow(r0)\}$. Thus, *even commonly used memory tests do not detect linked faults*.

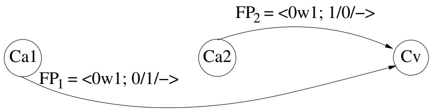

**Figure 2.** LF$_2$ = FP$_1$ $\rightarrow$ FP$_2$

### 3.2 Definition of linked faults

In practice, the vast majority of observed FPs are either two-cell FPs or single-cell FPs [2, 3, 4, 5, 6]. 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 FP$_1$ = $<S_1/F_1/R_1>$ is said to be **linked** to FP$_2$ = $<S_2/F_2/R_2>$ (denoted as FP$_1$ $\rightarrow$ FP$_2$) if the following three conditions are satisfied.

**1. Read operations of FP$_1$ and 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 consist of). This is because if one of the FPs is detected, then it will make no sense to talk about linked faults, since the fault is already detected. For example, RDF$_1$ = $<1r1/0/0>$ (see Table 1) cannot be linked to any other FP since this fault is immediately detected on the output.

**2. FP$_2$ masks 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. In the example of Figure 2, 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. FP$_2$ is compatible with FP$_1$.** This condition applies only in the case that the cell to which S$_2$ of the FP$_2$ should be applied, is 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, FP$_2$ = $<1r1/0/0>$ cannot be sensitized after FP$_1$ = $<0w1/0/->$, since after FP$_1$ the state of the v-cell is a faulty 0 which does not allow performing $1r1$ to sensitize FP$_2$.

## 4 Space of linked faults

This section starts with a classification of LFs, followed by enumerating the LFs in each class.

### 4.1 Classification of linked faults

State faults (SF) and state coupling faults (CFst) are faults sensitized by *sates* only (i.e., no operation is required by their $S$s). An analysis of the behavior of these faults reveals that they cannot be linked, because they are *dominant* [11]. Therefore, they are not further considered in this paper.

Since we consider FPs involving either one or two cells, LFs in this paper can involve *at most* three cells. Figure 3 shows the three classes of the LFs. Because the "linked relationship" is *not commutative*, the classification takes the order in which the two FPs are sensitized into consideration.

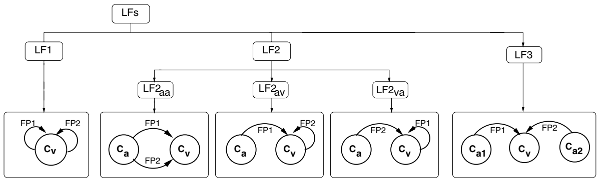

**Figure 3.** Classification of linked faults

**Table 3.** Space of single-cell linked faults

| 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$_0$: $<1w0/1/->$ | $\overline{\text{M}}$ | $\overline{\text{C}}$ | $\overline{\text{M}}$ | $\mathbf{L}_1$ | $\overline{\text{M}}$ | $\mathbf{L}_2$ | $\overline{\text{C}}$ | $\overline{\text{M}}$ | $\overline{\text{M}}$ | D |
| TF$_1$: $<0w1/0/->$ | $\overline{\text{C}}$ | $\overline{\text{M}}$ | $\mathbf{L}_3$ | $\overline{\text{M}}$ | $\mathbf{L}_4$ | $\overline{\text{M}}$ | $\overline{\text{M}}$ | $\overline{\text{C}}$ | D | $\overline{\text{M}}$ |
| WDF$_0$: $<0w0/1/->$ | $\overline{\text{M}}$ | $\overline{\text{C}}$ | $\overline{\text{M}}$ | $\mathbf{L}_5$ | $\overline{\text{M}}$ | $\mathbf{L}_6$ | $\overline{\text{C}}$ | $\overline{\text{M}}$ | $\overline{\text{M}}$ | D |
| WDF$_1$: $<1w1/0/->$ | $\overline{\text{C}}$ | $\overline{\text{M}}$ | $\mathbf{L}_7$ | $\overline{\text{M}}$ | $\mathbf{L}_8$ | $\overline{\text{M}}$ | $\overline{\text{M}}$ | $\overline{\text{C}}$ | D | $\overline{\text{M}}$ |
| 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$_0$: $<0r0/1/0>$ | $\overline{\text{M}}$ | $\overline{\text{C}}$ | $\overline{\text{M}}$ | $\mathbf{L}_9$ | $\overline{\text{M}}$ | $\mathbf{L}_{10}$ | $\overline{\text{C}}$ | $\overline{\text{M}}$ | $\overline{\text{M}}$ | D |
| DRDF$_1$: $<1r1/0/1>$ | $\overline{\text{C}}$ | $\overline{\text{M}}$ | $\mathbf{L}_{11}$ | $\overline{\text{M}}$ | $\mathbf{L}_{12}$ | $\overline{\text{M}}$ | $\overline{\text{M}}$ | $\overline{\text{C}}$ | D | $\overline{\text{M}}$ |

**1. LFs involving a single cell (LF1s)**: These are based on a combination of two single-cell FPs. Both FPs involved in the linked fault have the same a-cell as well as the same v-cell.

**2. 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 FPs. They are therefore divided into three types:

(a) 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. Each of the FPs forming LF2$_{aa}$ can be any of the FPs in Table 2.

(b) LF2$_{av}$: This LF is based on a combination of one two-cell FP$_1$ and one single-cell FP$_2$, where the two-cell FP is sensitized first. FP$_1$ can be any FP in Table 2, while FP$_2$ can be any FP in Table 1.

(c) LF2$_{va}$: This LF is similar to LF2$_{av}$, but here the single-cell FP should be sensitized first, followed by the two-cell FP. For these faults, test development is done in a different way than for LF2$_{av}$.

**3. LFs involving three cells (LF3s)**: These are based on a combination of two two-cell FPs with different a-cells, but the same v-cell.

In the remainder of this section, the domain of LF1s, LF2s and LF3s is presented.

## 4.2 Single-cell linked faults

Table 3 shows all single-cell FPs (see also Table 1) and the way they may be linked (i.e., whether FP$_1$ → FP$_2$), with the exception of state faults, since they are not considered. The FPs are listed both horizontally (FP$_1$) and vertically (FP$_2$), and each combination of FP$_1$ and FP$_2$ is given an entry in the table, indicating whether FP$_1$ can (or cannot) be linked to FP$_2$. There are four different entries used in the table, as explained next:

1. D: Condition 1 in the definition of linked faults is not satisfied (i.e., FP$_1$ or FP$_2$ is *detected*).
2. $\overline{\text{M}}$: Condition 2 is not satisfied (no *masking*).
3. $\overline{\text{C}}$: Condition 3 is not satisfied (no *compatibility*).
4. $\mathbf{L}_x$: FP$_1$ can be *linked* to FP$_2$ since all conditions in the definition of linked faults are satisfied.

The verification of FP$_1$→ FP$_2$ is done in such a 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 conditions. For example, when FPs are sensitized by a read operation, Condition 1 regarding detection is checked first. Note that the entries in Table 3 are not symmetrical with respect to the diagonal since the "*linked to*" relationship is not commutative. Table 3 shows that there are 12 LF1s, all of which are listed in Table 4.

**Table 4.** Instances of single-cell linked faults

| # | LF1 | # | LF1 |
|---|---|---|---|
| L$_1$ | TF$_0$ → WDF$_1$ | L$_2$ | TF$_0$ → RDF$_1$ |
| L$_3$ | TF$_1$ → WDF$_0$ | L$_4$ | TF$_1$ → RDF$_0$ |
| L$_5$ | WDF$_0$ → WDF$_1$ | L$_6$ | WDF$_0$ → RDF$_1$ |
| L$_7$ | WDF$_1$ → WDF$_0$ | L$_8$ | WDF$_1$ → RDF$_0$ |
| L$_9$ | DRDF$_0$ → WDF$_1$ | L$_{10}$ | DRDF$_0$ → RDF$_1$ |
| L$_{11}$ | DRDF$_1$ → WDF$_0$ | L$_{12}$ | DRDF$_1$ → RDF$_0$ |

## 4.3 Two-cell linked faults

Two-cell linked faults (LF2s) have been divided into three types (see Figure 3). The space of each type has been analyzed in a similar way as for LF1s, and the results show that:

1. There are 24 possible LF2$_{aa}$ faults given in Table 5. In the table, a compact notation is used for FPs (see Table 2). Note that L$_1$ and L$_4$ require $y_1 = x_2$ since they have to satisfy the compatibility condition. For example, CFds$_{x_1 O_1 y_1;0}$ → CFds$_{x_2 O_2 y_2;1}$ requires that $y_1 = x_2$ (i.e, the final state of the a-cell after performing $x_1 O_1 y_1$ should be the same as the initial state required by $x_2 O_2 y_2$), where $x, y \in \{0, 1\}$ and $O_1$, $O_2$ can be read or write operation.
2. There are 16 LF2$_{av}$ faults given in Table 6. An LF2$_{av}$ is based on a combination of one two-cell FP$_1$ and one single-cell FP$_2$.

**Table 5.** Instances of LF2$_{aa}$ faults

| # | LF2$_{aa}$ (FP$_1$→CFds) | # | LF2$_{aa}$ (FP$_1$→CFwd) | # | LF2$_{aa}$ (FP$_1$→CFrd) |
|---|---|---|---|---|---|
| L$_1$ | $\text{CFds}_{x_1 O_1 y_1;0} \rightarrow \text{CFds}_{x_2 O_2 y_2;1}$ | L$_9$ | $\text{CFds}_{xOy;0} \rightarrow \text{CFwd}_{y;1}$ | L$_{17}$ | $\text{CFds}_{xOy;0} \rightarrow \text{CFrd}_{y;1}$ |
| L$_2$ | $\text{CFds}_{x_1 O_1 y_1;1} \rightarrow \text{CFds}_{x_2 O_2 y_2;0}$ | L$_{10}$ | $\text{CFds}_{xOy;1} \rightarrow \text{CFwd}_{y;0}$ | L$_{18}$ | $\text{CFds}_{xOy;1} \rightarrow \text{CFrd}_{y;0}$ |
| L$_3$ | $\text{CFtr}_{x;0} \rightarrow \text{CFds}_{xOy;1}$ | L$_{11}$ | $\text{CFtr}_{x;0} \rightarrow \text{CFwd}_{x;1}$ | L$_{19}$ | $\text{CFtr}_{x;0} \rightarrow \text{CFrd}_{x;1}$ |
| L$_4$ | $\text{CFtr}_{x;1} \rightarrow \text{CFds}_{xOy;0}$ | L$_{12}$ | $\text{CFtr}_{x;1} \rightarrow \text{CFwd}_{x;0}$ | L$_{20}$ | $\text{CFtr}_{x;1} \rightarrow \text{CFrd}_{x;0}$ |
| L$_5$ | $\text{CFwd}_{x;0} \rightarrow \text{CFds}_{xOy;1}$ | L$_{13}$ | $\text{CFwd}_{x;0} \rightarrow \text{CFwd}_{x;1}$ | L$_{21}$ | $\text{CFwd}_{x;0} \rightarrow \text{CFrd}_{x;1}$ |
| L$_6$ | $\text{CFwd}_{x;1} \rightarrow \text{CFds}_{xOy;0}$ | L$_{14}$ | $\text{CFwd}_{x;1} \rightarrow \text{CFwd}_{x;0}$ | L$_{22}$ | $\text{CFwd}_{x;1} \rightarrow \text{CFrd}_{x;0}$ |
| L$_7$ | $\text{CFdr}_{x;0} \rightarrow \text{CFds}_{xOy;1}$ | L$_{15}$ | $\text{CFdr}_{x;0} \rightarrow \text{CFwd}_{x;1}$ | L$_{23}$ | $\text{CFdr}_{x;0} \rightarrow \text{CFrd}_{x;1}$ |
| L$_8$ | $\text{CFdr}_{x;1} \rightarrow \text{CFds}_{xOy;0}$ | L$_{16}$ | $\text{CFdr}_{x;1} \rightarrow \text{CFwd}_{x;0}$ | L$_{24}$ | $\text{CFdr}_{x;1} \rightarrow \text{CFrd}_{x;0}$ |

**Table 6.** Instances of LF2$_{av}$ faults

| # | LF2$_{av}$ (FP$_1$→WDF) | # | LF2$_{av}$ (FP$_1$→RDF) |
|---|---|---|---|
| L$_1$ | $\text{CFds}_{xOy;0} \rightarrow \text{WDF}_1$ | L$_9$ | $\text{CFds}_{xOy;0} \rightarrow \text{RDF}_1$ |
| L$_2$ | $\text{CFds}_{xOy;1} \rightarrow \text{WDF}_0$ | L$_{10}$ | $\text{CFds}_{xOy;1} \rightarrow \text{RDF}_0$ |
| L$_3$ | $\text{CFtr}_{x;0} \rightarrow \text{WDF}_1$ | L$_{11}$ | $\text{CFtr}_{x;0} \rightarrow \text{RDF}_1$ |
| L$_4$ | $\text{CFtr}_{x;1} \rightarrow \text{WDF}_0$ | L$_{12}$ | $\text{CFtr}_{x;1} \rightarrow \text{RDF}_0$ |
| L$_5$ | $\text{CFwd}_{x;0} \rightarrow \text{WDF}_1$ | L$_{13}$ | $\text{CFwd}_{x;0} \rightarrow \text{RDF}_1$ |
| L$_6$ | $\text{CFwd}_{x;1} \rightarrow \text{WDF}_0$ | L$_{14}$ | $\text{CFwd}_{x;1} \rightarrow \text{RDF}_0$ |
| L$_7$ | $\text{CFdr}_{x;0} \rightarrow \text{WDF}_1$ | L$_{15}$ | $\text{CFdr}_{x;0} \rightarrow \text{RDF}_1$ |
| L$_8$ | $\text{CFdr}_{x;1} \rightarrow \text{WDF}_0$ | L$_{16}$ | $\text{CFdr}_{x;1} \rightarrow \text{RDF}_0$ |

3. There are 18 LF2$_{va}$ faults given in Table 7. An LF2$_{va}$ is based on a combination of one single-cell FP$_1$ and one two-cell FP$_2$.

**Table 7.** Instances of LF2$_{va}$ faults

| LF2$_{va}$ (TF→FP$_2$) | | | |
|---|---|---|---|
| LF$_1$ | $\text{TF}_0 \rightarrow \text{CFds}_{xOy;1}$ | LF$_4$ | $\text{TF}_1 \rightarrow \text{CFds}_{xOy;0}$ |
| LF$_2$ | $\text{TF}_0 \rightarrow \text{CFwd}_{x;1}$ | LF$_5$ | $\text{TF}_1 \rightarrow \text{CFwd}_{x;0}$ |
| LF$_3$ | $\text{TF}_0 \rightarrow \text{CFrd}_{x;1}$ | LF$_6$ | $\text{TF}_1 \rightarrow \text{CFrd}_{x;0}$ |
| LF2$_{va}$ (WDF→FP$_2$) | | | |
| LF$_7$ | $\text{WDF}_0 \rightarrow \text{CFds}_{xOy;1}$ | LF$_{10}$ | $\text{WDF}_1 \rightarrow \text{CFds}_{xOy;0}$ |
| LF$_8$ | $\text{WDF}_0 \rightarrow \text{CFwd}_{x;1}$ | LF$_{11}$ | $\text{WDF}_1 \rightarrow \text{CFwd}_{x;0}$ |
| LF$_9$ | $\text{WDF}_0 \rightarrow \text{CFrd}_{x;1}$ | LF$_{12}$ | $\text{WDF}_1 \rightarrow \text{CFrd}_{x;0}$ |
| LF2$_{va}$ (DRDF→FP$_2$) | | | |
| LF$_{13}$ | $\text{DRDF}_0 \rightarrow \text{CFds}_{xOy;1}$ | LF$_{16}$ | $\text{DRDF}_1 \rightarrow \text{CFds}_{xOy;0}$ |
| LF$_{14}$ | $\text{DRDF}_0 \rightarrow \text{CFwd}_{x;1}$ | LF$_{17}$ | $\text{DRDF}_1 \rightarrow \text{CFwd}_{x;0}$ |
| LF$_{15}$ | $\text{DRDF}_0 \rightarrow \text{CFrd}_{x;1}$ | LF$_{18}$ | $\text{DRDF}_1 \rightarrow \text{CFrd}_{x;0}$ |

### 4.4 Three-cell linked faults

Three-cell linked faults describe linking two two-cell FPs with different a-cells and the same v-cell (see Figure 3). The instances of this class of linked faults are exactly the same as those for two-cell LF2$_{aa}$ (see Table 5). The only difference is that the compatibility condition does not apply for LF3s since both a-cells are different. For example, CFds$_{0w1;1}$ can be linked to CFds$_{0r0;0}$ if LF3 is considered but not if LF2$_{aa}$ is considered.

### 4.5 Validation of linked faults

In the literature [2, 3, 4, 5, 6], all possible resistive defects of the memory cell array for SRAMs and DRAMs have been inserted and simulated at the electrical level using real designs, where only a single defect is inserted and simulated at a time. The electrical faults observed have been described as FPs. All FPs considered for LFs in the previous section have been shown to exist. If only one defect at a time is considered, LFs can only take place in case a single defect causes two different FPs with the *opposite* fault effect. Simulation shows that there are no such LFs for SRAMs, while these are very limited for DRAMs [3, 4, 5, 6]. Since LFs are based on a combination of two FPs, the analysis of such faults has to consider *two defects at a time*. Each defect can then sensitize a FP which has an opposite fault effect with respect to the other one, such that masking may take place. Note that the concept of linked faults involves only two faults at a time (see definition in Section 3), which means when three or more faults are present, only two of them can be linked at a time. In other words, no matter how many defects are present in the memory, the linked fault analysis presented in this paper can detect them. Next, an example is presented based on defect injection and simulation where a defective circuit shows linked faulty behavior.

**Example of LF1:** An LF1 example, TF$_0$ →RDF$_1$, is shown in Figure 4 for SRAMs. The resistive defect D1 causes TF$_0$ $=<1w0/1/->$, while the resistive defect D2 causes RDF$_1$ $=<1r1/0/->$ in case the resistance of the two defects belong to certain ranges [2, 5]. If now the following 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 ($r$). The linked fault TF$_0$ →RDF$_1$ will result. The linking process starts by performing $1w0$ on a cell, which fails to set the cell to 0. 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 $r1$ results *internally* in a fault, performing both operations as an *external* sequence does not result in any fault in the memory. Therefore, a proper test for this faulty behavior should sensitize and detect only one of the two faults, without sensitizing the other. Such a test is described in the next section.

## 5 Test design for linked faults

This section shows first that the traditional tests do not necessary cover LFs. Thereafter, it introduces a methodology to be used in order to design tests for LFs. Due to the lack of space in this paper, only LF1s are treated here (see Table 4).

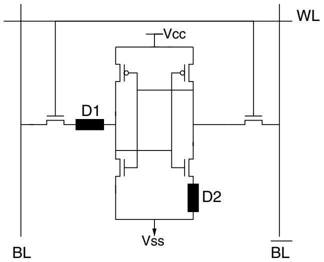

**Figure 4.** Defects causing TF$_0$ → RDF$_1$

## 5.1 Fault coverage of traditional tests

Table 8 summarizes the fault coverage (FC) of the most known traditional tests for LFs. The table shows for example that March C- detects 8 of 12 LF1 faults of Table 4, 10 of 24 LF2$_{aa}$ faults of Table 5, 11 of 16 LF2$_{av}$ faults of Table 6, 9 of 18 LF2$_{va}$ faults of Table 7, and 10 of 24 LF3 faults. The table clearly shows that the traditional tests designed for *simple* faults do not cover all LFs. The best test achieving the highest FC for LFs are March SR and PMOVI.

**Table 8.** Linked fault coverage for traditional tests

| Tests | LF1 | LF2$_{aa}$ | LF2$_{av}$ | LF2$_{va}$ | LF3 | FC |
|---|---|---|---|---|---|---|
| Scan | 8/12 | 4/24 | 8/16 | 3/18 | 4/24 | 27/94 |
| MATS+ | 8/12 | 0/24 | 8/16 | 1/18 | 0/24 | 17/94 |
| MATS++ | 10/12 | 0/24 | 8/16 | 4/18 | 0/24 | 22/94 |
| March C- | 10/12 | 10/24 | 11/16 | 9/18 | 10/24 | 50/94 |
| PMOVI | 10/12 | 15/24 | 11/16 | 12/18 | 15/24 | 63/94 |
| March SR | 10/12 | 16/24 | 13/16 | 15/18 | 18/24 | 72/94 |
| March G | 10/12 | 9/24 | 10/16 | 13/18 | 9/24 | 51/94 |
| Walk 1/0 | 8/12 | 12/24 | 9/16 | 10/18 | 12/24 | 51/94 |
| Galpat | 10/12 | 13/24 | 9/16 | 14/18 | 13/24 | 59/94 |

## 5.2 Tests for single-cell linked faults

This section discusses systematically designing a test for LF1s. First, the needed detection conditions for LF1s are presented, and then they are compiled into a march test.

### 5.2.1 Detection conditions

Detection condition for LF1=FP$_1$→FP$_2$ faults are given based on the idea that any LF1 is detectable when at least one of the FPs forming LF1 can be sensitized and detected *in isolation* (i.e., without allowing the other FP to mask the fault).

**A. Detection condition for FP$_1$→WDF$_1$**

The LF1, FP$_1$→WDF$_1$, where FP$_1$∈ {TF$_0$, WDF$_0$, DRDF$_0$} (i.e., L$_1$, L$_5$ and L$_9$ of Table 4) 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 L$_1$, L$_5$ and L$_9$.

For *simple* WDF$_1$ $=<1w1/0/->$, 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 that contains the four march elements: $\Updownarrow(...,O0)$; $\Updownarrow(w1)$; $\Updownarrow(w1)$; $\Updownarrow(r1,...)$, which can be merged into one, two, or three elements.

The fault is first sensitized by applying a non-transition 'w1' operation (i.e., applying a 'w1' to the cell which contains 1). The '$O$' denotes any operation, and '$O0$' guarantees that the content of the accessed cell is 0 before the first 'w1' operation is applied, such that the first write operation will be an up transition and the second 'w1' operation will be a non-transition 'w1' operation which is required to sensitize the fault. The fault will be then detected by 'r1' operation, which will return a wrong value 0 rather than the expected value 1. In the absence of '$O0$', the fault may not be detected. E.g., if before performing the first 'w1' operation, the content of the cell was 1, then the first 'w1' will be an non-transition write operation and therefore will cause the cell to flip to 0 due to WDF$_1$ $=<1w1/0/->$, 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.

For *linked* 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 4). Masking will take place and the fault will be not detected.
2. A non-transition 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.

By allowing '$O0$' to be '$r0$', the TF$_0$ →WDF$_1$ and the WDF$_0$ →WDF$_1$ will be detected by detecting WDF$_1$ in isolation and using the same test condition as above, since '$r0$' will not sensitize TF$_0$ neither WDF$_0$. However, DRDF$_0$ →WDF$_1$ will be not detected since the '$r0$' can sensitize a DRDF$_0$ that can be linked to WDF$_1$. Adding an extra '$r0$' after $\Updownarrow(...,r0)$ will detect DRDF$_0$ →WDF$_1$ by detecting the DRDF$_0$ in isolation. Therefore, in order to detect all FP$_1$→WDF$_1$, where FP$_1$∈ {TF$_0$, WDF$_0$, DRDF$_0$}, the following condition is needed.

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

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

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

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

**B. Detection condition for $\text{FP}_1 \rightarrow \text{RDF}_1$**

The LF1s, $\text{FP}_1 \rightarrow \text{RDF}_1$, where $\text{FP}_1 \in \{\text{TF}_0, \text{WDF}_0, \text{DRDF}_0\}$ (i.e., $\text{L}_2$, $\text{L}_6$ and $\text{L}_{10}$) are detectable if the $\text{RDF}_0$ is sensitized and detected in isolation.

*Condition lRDF$_1$*: The $\text{RDF}_1$ is detectable in isolation by a march test if the test contains the march element: $\Updownarrow(..., r1, ...)$. Applying a 'r1' operation will flip the cell to 0 and return the incorrect value 0.

Any operation performed before the 'r1' operation of $\Updownarrow(..., r1, ...)$ can be a 'r1', a transition 'w1' operation ('0w1'), or a non-transition write operation ('1w1'). By inspecting $\text{FP}_1$ in $\text{FP}_1 \rightarrow \text{RDF}_1$ (see Table 4), one can see that there is no FP that can be linked to $\text{RDF}_1$, and sensitized by '1r1', '0w1', or '1w1'. Therefore there is no possible masking, and $\Updownarrow(..., r1, ...)$ will detect all $\text{FP}_1 \rightarrow \text{RDF}_1$ of Table 4.

**C. Detection condition for all LF1s**

In a similar way as above, detection conditions for FPs linked to $\text{WDF}_0$ and to $\text{RDF}_0$ (see Table 4) have been developed . All these detection conditions have been merged into the following condition:
*Condition LF1:* Any LF1 of Table 4 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, two, three, four, five, six, seven, eight or nine elements.

- Case A: to detect $\text{L}_1$, $\text{L}_2$, $\text{L}_5$, $\text{L}_6$ , $\text{L}_9$ and $\text{L}_{10}$
  $\Updownarrow(..., r0); \Updownarrow(r0); \Updownarrow(w1); \Updownarrow(w1); \Updownarrow(r1, ...)$
- Case B: to detect $\text{L}_3$, $\text{L}_4$, $\text{L}_7$, $\text{L}_8$ , $\text{L}_{11}$ and $\text{L}_{12}$
  $\Updownarrow(..., r1); \Updownarrow(r1); \Updownarrow(w0); \Updownarrow(w0); \Updownarrow(r0, ...)$

### 5.2.2 Test for single-cell linked faults

The test detecting all LF1s is shown in Figure 5, 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., $\text{M}_1$) of the test contains the five march elements of Case A, while $\text{M}_2$ contains the five march elements of Case B. Note that the first read operation of $\text{M}_2$ can be removed without having any impact on the fault coverage of the test. $\text{M}_1$ will then contain the first march element of Case B, and $\text{M}_2$ 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.

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

**Figure 5.** March LF1

# 6 Conclusions

In this paper, a complete analysis of linked faults in RAMs has been presented based on the concept of fault primitives. A precise definition of LFs was introduced and used to establish the whole space of LFs. It has been shown that traditional tests designed for simple faults do not necessarily cover LFs. In addition, a methodology to design the appropriate LF tests was introduced, where detection conditions are first developed, and then compiled into tests. This methodology has been applied to derive the March LF1 test that detects all single-cell linked faults. It is interesting to note that the concept of linked faults involves only two faults at a time, which means when three or more faults are present, only two of them can be linked at a time. In other words, no matter how many defects are present in the memory, the linked fault analysis presented in this paper can detect them.

# References

[1] M.S. Abadir and J.K. Reghbati, "Functional Testing of Semiconductor Random Access Memories," *ACM Computer Surveys*, 15(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.

[3] Z. Al-Ars and Ad J. van de Goor, "Impact of Memory Cell Array Bridges on the Faulty Behavior in Embedded DRAMs," *in Proc. Asian Test Symp.*, pp. 282–289, 2000.

[4] R. Dekker, F. Beenaker, and L. Thijssen, " A Realistic Fault Model and Test Algorithm for Static Random Access Memories," *in IEEE Trans. on CAD*, vol. 9, no. 6, pp. 567–572, 1990.

[5] S. Hamdioui and A.J. van de Goor, "Experimental Analysis of Spot Defects in SRAMs: Realistic Fault Models and Tests," *in Proc. Asian Test Symp.*, pp. 131–138, 2000.

[6] V.-K. Kim and T. Chen, "On Comparing Functional Fault Coverage and Defect Coverage for Memory Testing," *in IEEE Trans. on CAD*, vol. 18, no. 11, pp. 1676–1683, 1999.

[7] M. Marinescu, "Simple and Efficient Algorithms for Functional RAM Testing," *in Proc. IEEE Int'l Test Conf.*, pp. 236–239, 1982.

[8] C.A. Papachristou and N.B. Saghal, "An Improved Method for Detecting Functional Faults in Semiconductor Random Access Memories," *in IEEE Trans. on Computers*, vol. C-34, no. 2, pp. 110–116, 1985.

[9] A.J. van de Goor and Z. Al-Ars, "Functional Fault Models: A Formal Notation and Taxonomy," *in Proc. IEEE VLSI Test Symp.*, pp. 281–289, 2000.

[10] A.J. van de Goor and A. Paalvast, "Industrial Evaluation of DRAM SIMM Tests," *in Proc. IEEE Int'l Test Conf.*, pp. 426–435, 2000.

[11] A.J. van de Goor, *et. al.*, "March LR: A Test for Realistic Linked Faults," *in Proc. IEEE VLSI Test Symp.*, pp. 272–280, 1996.