Efficient Tests for Realistic Faults in Dual-Port SRAMs

Said Hamdioui, *Associate Member*, *IEEE*, and Ad J. van de Goor, *Fellow*, *IEEE*

Abstract—This paper begins with an overview of realistic fault models for dual-port memories, divided into single-port faults and faults unique for dual-port memories. The latter faults cannot be detected with the conventional single-port memory tests; they require special tests. A precise notation for all faults, such that ambiguities and misunderstandings will be prevented, has been emphasized. Next, the paper presents a methodology to design tests for realistic unique dual-port memory faults, resulting in a set of three linear single-addressing tests which are merged into a single march test (March s2PF), and one linear double-addressing test (March d2PF). March s2PF and March d2PF have been implemented at Intel. The results show that they detect unique faults, i.e., faults that cannot be detected with conventional single-port memory tests. This make them very attractive industrially.

Index Terms—Multiport/single-port memories, fault models, weak faults, march tests, fault coverage.

1 Introduction

Fast and efficient testing is an important step in any manufacturing process. With the recent advances in semiconductor technology, the design and use of memories for realizing complex systems-on-a-chip has been widespread. The cost of testing such memories increases rapidly with every generation [1]. Precise and realistic fault modeling, and efficient test design, in order to keep test cost and time within economically acceptable limits, are therefore essential.

A novel characteristic of today's memories is the presence of multiple ports to allow the two common operations (read and/or write) to be performed simultaneously. Testing of such *Multiport (MP)* memories requires special tests since the multiple and simultaneous access can sensitize faults that are different from the conventional *single-port (SP)* memory faults.

In spite of the growing use of MP memories, little *experimental* work has been published. In [2], an ad hoc test with no specific fault model was described. In [3], a BIST circuit, based on a serial interfacing technique for embedded dual-port(i.e, *two-port (2P)*) memories, was reported. However, the used fault models were very *simplistic*, and the proposed BIST requires a modification of the design. For the same fault models, modified march tests and BIST circuits were reported in [4], [5], [6]. In [7], [8], [9], [10], it has been shown *theoretically* that the conventional tests for SP memories are *insufficient* for MP memories. Moreover, *theoretical* fault models, together with their tests were developed. However, the introduced fault models are *not* based on any experimental/industrial analysis. In addition, the proposed tests have a time complexity which is exponentially proportional with the number of ports of the MP memory; that makes them not practical. In [11], port interferences in 2P memories were *experimentally* analyzed, based on an industrial design and SPICE simulation; however, the analysis was restricted to only the interference between the two ports. A similar, but theoretical work, has been reported in [12].

It can be seen from the above that *realistic* fault models and *optimal* tests still have to be established. This paper deals with that subject and is organized as follows: Section 2 gives an overview of realistic fault models for 2P memory cell arrays, as established by [13]. Section 3 establishes conditions for detecting two-port faults. Section 4 derives optimal march tests for such faults and classifies them. Section 5 presents the test strategy. Section 6 presents industrial test results, while Section 7 ends with conclusions.

2 Fault Models for 2P-SRAMs

Fig. 1 shows a differential 2P memory cell that will be the subject of this paper. It has two *read-write* ports, i.e., both ports can be used for read as well as for write operations. The following operations are allowed to be performed:

- Single read as well as write operations through each port.
- Two simultaneous read operations to the same location, as well as to different locations.
- Two simultaneous write operations to different locations.
- Simultaneous read and write to different locations.
- Simultaneous read and write to the same locations. However, in that case, the read data will be discarded, i.e, the write operation has a high priority.

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

- *The authors are with Delft University of Technology, Faculty of Information Technology and Systems, Laboratory of Computer Engineering, Mekelweg 4, 2628 CD Delft, The Netherlands.*
E-mail: s.hamdioui@its.tudelft.nl and vdGoor@ce.et.tudelft.nl.

Manuscript received 23 Jan 2001; revised 15 May 2001; accepted 18 July 2001.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 113521.

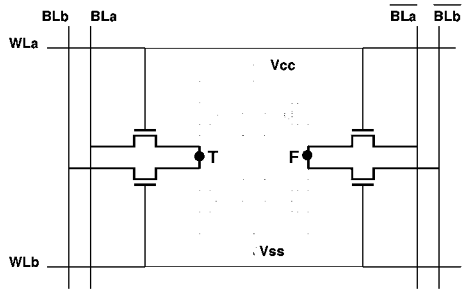


Fig. 1. Differential two-port memory cell.

three groups: *opens, shorts,* and *bridges*; whereby an open is an extra resistance within a connection, a short is an undesired resistive path between a node and V_{cc} or V_{ss}, while a bridge is an undesired resistive path between two connections, which are not V_{cc} or V_{ss}.

In [13], a complete experimental analysis for opens, shorts, and bridges in a differential 2P memory cell has been done, similar to that done for single-port SRAMs [14]. The SPICE simulation of all possible SDs at the electrical level of the memory cell has been performed by examining the resistance range of each SD from 0 to ∞, e.g., for shorts, the first simulation has been done for the case that $R_{sh} = 0\Omega$. If no fault occurs in this case, then it does not make sense to simulate other cases for bigger values of R_{sh}. The boundary between the different behaviors of the cell (i.e., proper operation, faulty behavior), is searched by stepping through the resistor value range. This stepping is started for shorts by $R_{sh} = 10^k\Omega$ in which k is increased by 1 each time (starting by $k = 0$). If a transition between two different behaviors occurs, then the region between 10^k and 10^{k+1} for shorts is examined with a step size of 10^k. These actions are repeated until the boundary has been found with the desired accuracy. The used simulation methodology verifies all allowed operations in the analyzed 2P memory. During the simulation, any faulty behavior (in the presence of a certain SD) is reported in terms of a *fault primitive (FP)*. An FP is a compact notation describing the fault. The simulation results show that the sensitized FPs are *strongly dependent* on the resistance value of the defect.

The faults caused by the SDs, and expressed in terms of FPs, are translated into *functional fault models (FFMs)*. These are used to design tests, e.g., a stuck at fault (SAF) is a FFM, while the MATS+ [15] test has been designed to detect SAFs. An FFM is defined as a nonempty set of FPs. This section gives FFMs based on FPs, as established by [13], [14]. The FFMs for 2P memories are divided into single-port FFMs and two-port FFMs.

2.1 Single-Port FFMs

Single-port FFMs (abbreviated as $1PFs$) are divided into faults involving a *single-cell (1PF1s)* and faults involving *two-cell (1PF2s)*; see Fig. 2. The 1PF1s consist of single-cell FPs; they have the property that the cell used for sensitizing the fault is the same cell as where the fault appears. The 1PF2s have the property that: a) the application of a single-port operation (solid arrow in Fig. 2) to the *aggressor cell (c_a)*, b) the state of the cell c_a (dashed arrow in the figure), or c) the application of a single-port operation to *the victim cell* (c_v) with cell c_a in certain state, has as a consequence that a fault will be sensitized in the cell c_v.

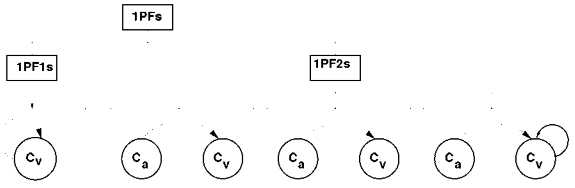


Fig. 2. Classification of 1PFs.

Before listing the 1PFs, a precise compact notation, referred as a *fault primitive (FP)*, which will prevent ambiguities and misunderstandings, will be introduced.

- $< S/F/R >$ (or $< S/F/R >_v$): denotes an FP involving a single-cell; the cell c_v (victim cell) used to sensitize a fault is the same as where the fault appears. S describes the value/operation *sensitizing* the fault; $S \in \{0, 1, w0, w1, w\uparrow, w\downarrow, r0, r1, \forall\}$, whereby 0 (1) denotes a *zero* (*one*) value, $w0$ ($w1$) denotes a write 0 (1) operation, $w\uparrow$ ($w\downarrow$) denotes an up (down) transition write operation, $r0$ ($r1$) denotes a read 0 (1) operation, and $\forall$ denotes any operation ($\forall \in \{0, 1, w1, w0, w\uparrow, w\downarrow, r1, r0\}$). If the fault effect of S appears after a time T, then the sensitizing operation is given as S_T.
- $< S_a; S_v/F/R >$ (or $< S_a; S_v/F/R >_{a,v}$): Denotes an FP involving two cells; S_a describes the sensitizing operation or state of the *aggressor cell (a-cell)*; while S_v describes the sensitizing operation or state of the *victim cell (v-cell)*. The a-cell (c_a) is the cell sensitizing a fault in an other cell called the v-cell (c_v). The set S_i is defined as:

$$S_i \in \{0, 1, w0, w1, w\uparrow, w\downarrow, r0, r1\}(i \in \{a, v\}).$$

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

2.1.1 The 1PF1 Fault Subclass

The 1PF1 faults are FFMs consisting of single-port, single-cell FPs. They consist of nine FFMs [13], [14], and are listed in Table 1. The first column in the table gives the abbreviation of the FFM, while the second column shows the FPs the FFM consists of. The nine 1PF1s are:

1. *Stuck-at Fault (SAF)*: The logic value of a cell is always "0" or "1." The SAF consists of two FPs: $< \forall/0/- >$, and $< \forall/1/- >$.

TABLE 1
List of 1PF1s; $x \in \{0,1\}$

FFM	Fault primitives
SAF	$< \forall/0/- >$, $< \forall/1/- >$
TF	$< w\uparrow/0/- >$, $< w\downarrow/1/- >$
RDF	$< r0/\uparrow/1 >$, $< r1/\downarrow/0 >$
DRDF	$< r0/\uparrow/0 >$, $< r1/\downarrow/1 >$
IRF	$< r0/0/1 >$, $< r1/1/0 >$
RRF	$< r0/0/? >$, $< r1/1/? >$
DRF	$< 1_T/\downarrow/- >$, $< 0_T/\uparrow/- >$ $< x_T/?/- >$
NAF	$< w\uparrow/0/- >$, $< w\downarrow/1/- >$, $< rx/x/? >$
USF	$< wx/?/- >$, $< rx/?/? >$

2. *Transition Fault (TF)*
3. *Read Destructive Fault (RDF)* [18]
4. *Deceptive Read Destructive Fault (DRDF)* [18]
5. *Incorrect Read Fault (IRF)*
6. *Random Read Fault (RRF)*
7. *Data Retention Fault (DRF)* [19]
8. *No Access Fault (NAF)*
9. *Undefined State Fault (USF)*

2.1.2 The 1PF2 Fault Fault Subclass

The 1PF2 faults are FFMs consisting of single-port FPs, which involve two cells. They consist of seven FFMs [13], [14]; see Table 2.

1. *Disturb Coupling Fault* (CF_{ds}) [17]: A *Disturb Coupling Fault* is defined as a fault whereby the v-cell undergoes a transition due to a write or a read operation applied to the a-cell. It consists of eight FPs.
2. *State Coupling Fault* (CF_{st}) [19]
3. *Incorrect Read Coupling Fault* (CF_{ir})
4. *Random Read Coupling Fault* (CF_{rr})
5. *Deceptive Read Destructive Coupling Fault* (CF_{dr})
6. *Read Destructive Coupling Fault* (CF_{rd})
7. *Transition Coupling Fault* (CF_{tr})

2.2 Two-Port FFMs

In the representation of the 2PFs, the following terminology will be reintroduced [7], [8], [9], [10]:

TABLE 2
List of 1PF2s; $x \in \{0,1\}$

FFM	Fault primitives
CF$_{ds}$	$< wx;\ 0/\uparrow/- >$, $< wx;\ 1/\downarrow/- >$, $< rx;\ 0/\uparrow/- >$, $< rx;\ 1/\downarrow/- >$
CF$_{st}$	$< 1;\ 1/0/- >$, $< 1;\ 0/1/- >$, $< 0;\ 1/0/- >$, $< 0;\ 0/1/- >$
CF$_{ir}$	$< 0;\ r0/0/1 >$, $< 0;\ r1/1/0 >$, $< 1;\ r0/0/1 >$, $< 1;\ r1/1/0 >$
CF$_{rr}$	$< 0;\ r0/0/? >$, $< 0;\ r1/1/? >$, $< 1;\ r0/0/? >$, $< 1;\ r1/1/? >$
CF$_{dr}$	$< 0;\ r0/\uparrow/0 >$, $< 0;\ r1/\downarrow/1 >$, $< 1;\ r0/\uparrow/0 >$, $< 1;\ r1/\downarrow/1 >$,
CF$_{rd}$	$< 0;\ r0/\uparrow/1 >$, $< 0;\ r1/\downarrow/0 >$, $< 1;\ r0/\uparrow/1 >$, $< 1;\ r1/\downarrow/0 >$
CF$_{tr}$	$< 0;\ w\downarrow/1/- >$, $< 0;\ w\uparrow/0/- >$, $< 1;\ w\downarrow/1/- >$, $< 1;\ w\uparrow/0/- >$

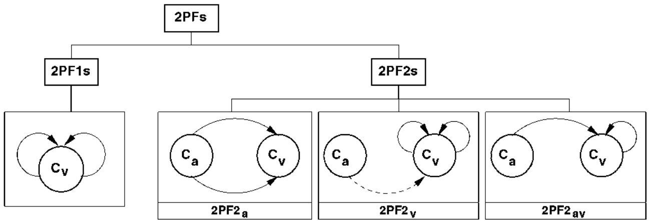


Fig. 3. Classification of 2PFs.

- *Strong Fault:* This is a memory fault that can be **fully** *sensitized* by an operation, e.g., an SP write or read operation fails, two simultaneous read operations fail, etc. That means that the state of the v-cell is incorrectly changed, cannot be changed, or that the sense amplifier(s) return(s) an incorrect results.
- *Weak Fault:* This is a fault which is **partially** sensitized by an operation, e.g., due to a defect that creates a small disturbance of the voltage of the true node of the cell. However, a fault can be *fully sensitized* (i.e., becomes strong) when two (or more) weak faults are sensitized simultaneously since their fault effects can be additive. This may occur when a 2P operation is applied. Note that in the presence of a weak fault, all SP (read and write) operations pass correctly and that the 2P operations may pass correctly. The latter will be the case if the fault effects of the weak faults are not sufficient enough to fully sensitize a fault.

Two-port faults (2PFs) cannot be sensitized using SP operations; they require the use of the two ports simultaneously. The 2PFs can be considered as a combination of *two weak faults*. The fault effects of two or more weak faults may be additive and, hence, fully sensitize a fault when the weak faults are sensitized simultaneously.

2PFs can be divided into *faults involving a single cell (2PF1s)* and *faults involving two cells (2PF2s)* [13]; see Fig. 3.

The 2PF1s are based on a combination of *two single-cell weak faults*. In addition, the two a-cells are the same as the v-cell. In order to sensitize a 2PF1, the same cell has to be acted upon simultaneously via the two ports. To denote a 2PF1 fault, the following notation will be used:

$< S_1 : S_2/F/R >_v$. It denotes a two-port FP involving a single cell (v-cell). This FP requires the use of the two ports *simultaneously*. S_1 and S_2 describe the sensitizing operations or states of the cell; ":" denotes the fact that S_1 and S_2 are applied *simultaneously* through the two ports. F describes the value of the v-cell. Note that the sensitizing operations are applied to the same cell as where the fault appears. R is the read result of S_1 (and/of S_2) if it is a read operation.

The 2PF2s are based on a combination of weak single-cell faults and weak faults involving two cells. Depending on to which cells the two simultaneous operations are applied (to the a-cell and/or to the v-cell), the 2PF2s are divided into three types (see Fig. 3):

1. *The* $2PF2_a$: This fault is sensitized in cell c_v by applying two simultaneous operations to the same *a-cell* c_a (solid arrows in the Fig. 3). Note that, in this case, the 2PF is a combination of two weak faults

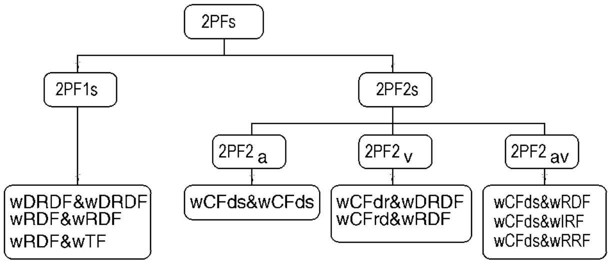


Fig. 4. A taxonomy of 2PFs.

involving two cells; both weak faults have the same a-cell as well as the same v-cell. To denote a $2PF2_a$, the following notation is used:

$< S_a : S_a; S_v/F/R >_{a,v}$. It denotes an FP whereby both sensitizing operations, S_a, are applied simultaneously to the a-cell. S_v denotes the state of the v-cell. F denotes the value of the faulty cell c_v. Note that, in that case, R will be replaced with "$-$" since S_v cannot be a read operation.

2. *The* $2PF2_v$: This fault is sensitized in cell c_v by applying two simultaneous operations to the same cell c_v (solid arrows in Fig. 3), while the a-cell has to be in certain state (dashed arrow in the figure). Note that this fault is a combination of two weak faults: a single-cell weak fault and a weak weak fault involving two cells whereby the operation has to be performed to the v-cell while the a-cell has to be in a certain state. To denote a $2PF2_v$, the following notation is used:

   $< S_a; S_v : S_v/F/R >_{a,v}$. It denotes an FP whereby both sensitizing operations, S_v, are applied simultaneously to the v-cell. S_a describes the state of the a-cell.
3. *The* $2PF2_{av}$: This fault is sensitized by applying two simultaneous operations: one to cell c_a and one to cell c_v; see Fig. 3. It is a combination of a single-cell weak fault and a weak fault involving two cells. Such fault is denoted as follows:

   $< S_a : S_v/F/R >_{a,v}$. It denotes an FP whereby the sensitizing operation S_a is applied to the a-cell simultaneously with the sensitizing operation S_v is applied to the v-cell.

In addition to the above introduced notation for FPs, the following notation will be used for describing 2PFs.

- $< fault_1 > \& < fault_2 >$: denotes a 2PF consisting of two weak faults; "&" denotes the fact that the two faults *in parallel* (i.e., simultaneously) form the 2PF.
- F denotes a *strong fault* F, while wF denotes the *weak fault* F. For example, RDF denotes a strong Read Destructive Fault, while $wRDF$ denotes a weak Read Destructive Fault.

A taxonomy of all realistic 2PFs for the 2P memory cell array is given in Fig. 4, while Table 3 shows the FPs of which each 2PF is composed. These 2PFs will be explained in detail in the following sections.

2.2.1 The 2PF1 Fault Subclass

The 2PF1 fault class consists of single-cell FPs, involving two simultaneous operations in order to be sensitized. They consist of three FFMs [13]:

TABLE 3
List of 2PFs; $x \in \{0,1\}$ and $d = \mathrm{don't\ care}$

FFM	Fault primitives
wDRDF&wDRDF	$< r0 : r0/\uparrow/0 >, < r1 : r1/\downarrow/1 >$
wRDF&wRDF	$< r0 : r0/\uparrow/1 >, < r1 : r1/\downarrow/0 >$
wRDF&wTF	$< r0 : w\uparrow/0/- >, < r1 : w\downarrow/1/- >$
wCF$_{ds}$&wCF$_{ds}$	$< w0 : rd; 0/\uparrow/- >, < w0 : rd; 1/\downarrow/- >$ $< w1 : rd; 0/\uparrow/- >, < w1 : rd; 1/\downarrow/- >$ $< rx : rx; 0/\uparrow/- >, < rx : rx; 1/\downarrow/- >$
wCF$_{dr}$&wDRDF	$< 0; r0 : r0/\uparrow/0 >, < 0; r1 : r1/\downarrow/1 >,$ $< 1; r0 : r0/\uparrow/0 >, < 1; r1 : r1/\downarrow/1 >,$
wCF$_{rd}$&wRDF	$< 0; r0 : r0/\uparrow/1 >, < 0; r1 : r1/\downarrow/0 >,$ $< 1; r0 : r0/\uparrow/1 >, < 1; r1 : r1/\downarrow/0 >$
wCF$_{ds}$&wRDF	$< w0 : r0/\uparrow/1 >, < w0 : r1/\downarrow/0 >,$ $< w1 : r0/\uparrow/1 >, < w1 : r1/\downarrow/0 >$
wCF$_{ds}$&wIRF	$< w0 : r0/0/1 >, < w0 : r1/1/0 >,$ $< w1 : r0/0/1 >, < w1 : r1/1/0 >$
wCF$_{ds}$&wRRF	$< w0 : r0/0/? >, < w0 : r1/1/? >,$ $< w1 : r0/0/? >, < w1 : r1/1/? >$

1. $wDRDF\&wDRDF$: Applying two simultaneous read operations to a single cell causes the cell to flip; while the sense amplifiers return the *correct values*; see Fig. 3. This is because the flipping of the cell happens relatively slowly. In order to detect this fault, at least an extra single read operation has to be performed to the same cell. The wDRDF&wDRDF consists of two FPs: $< r0 : r0/\uparrow/0 >_v$ (i.e., applying two simultaneous $r0$ operations to cell c_v will flip the cell to 1, and the sense amplifiers return the correct values), and $< r1 : r1/\downarrow/1 >_v$. It can be caused by the following defects [13] a) drain/source of the pull-down transistor of the cell broken, b) true or false node shorted to V_{ss}, and c) short between a cell's node and a word line of an adjacent cell.
2. $wRDF\&wRDF$: Applying two simultaneous read operations to a single cell causes the cell to flip and the sense amplifiers return *incorrect* values. The wRDF&wRDF consists of two FPs: $< r0 : r0/\uparrow/1 >_v$ and $< r1 : r1/\downarrow/0 >_v$. It can be caused by the same defects as those causing the wDRDF&wDRDF, but with different resistance values of the defect.
3. $wRDF\&wTF$: A cell fails to undergo a write transition if a read operation is applied to the same cell simultaneously. The wRDF&wTF consists of two FPs: $< r0 : w\uparrow/0/- >_v$ and $< r1 : w\downarrow/1/- >_v$. It can be caused by bridges between bit lines belonging to the same column, to different ports and to different sides (i.e., true side and false side) of the cell [13].

2.2.2 The 2PF2 Fault Subclass

The 2PF2s consist of FPs involving two cells and require two simultaneous operations in order to be sensitized; they are divided into three types [13].

The $\mathbf{2PF2_a}$: This type consists only of one FFM: $wCF_{ds}\&wCF_{ds}$. Applying two simultaneous operations to cell c_a will sensitize a fault in cell c_v, i.e., cell c_v flips; see Fig. 3. The $wCF_{ds}\&wCF_{ds}$ consists of eight FPs: $< w0 : \mathrm{r}d; 0/\uparrow/- >$, $< w0 : \mathrm{r}d;\ 1/\downarrow/- >$, $< w1 : \mathrm{r}d;\ 0/\uparrow/- >$, $< w1 : \mathrm{r}d;\ 1/\downarrow/- >$, $< \mathrm{r}x : \mathrm{r}x;\ 0/\uparrow/- >$, and $< \mathrm{r}x : \mathrm{r}x; 1/\downarrow/- >$; whereby $x \in \{0,1\}$, and d denotes the *don't care value*. Note that the $< w0 : \mathrm{r}d;\ 1/\downarrow/- >$ denotes only one

FP since the read value is irrelevant; the read operation is used to sensitize the fault. Note also that $< \mathrm{r}x : \mathrm{r}x;\ 0/\uparrow/->$ denotes two FPs since $x \in \{0,1\}$. The 2PF2$_a$ can be caused by bridges between nodes of adjacent cells belonging to the same row, to the same column, or on same diagonal. They can also be caused by a bridge between a node of a cell and a bit line of an adjacent cell in the same row [13].

The 2PF2$_\mathrm{v}$: This type can be caused by bridges between nodes of adjacent cells belonging to the same row, the same column, or on the same diagonal [13]. It consists of two FFMs:

1. *wCF$_{dr}$&wDRDF*: Applying two simultaneous read operations to cell c_v will cause the cell to flip if cell c_a is in a certain state. The read operations return *correct* values. The wCF$_{dr}$&wDRDF consists of four FPs: $<0;\ r0:r0/\uparrow/0>$, $<0;\ r1:r1/\downarrow/1>$, $<1;\ r0:r0/\uparrow/0>$, and $<1;\ r1:r1/\downarrow/1>$.
2. *wCF$_{rd}$&wRDF*: Applying two simultaneous read operations to cell c_v will cause the cell to flip if cell c_a is in a certain state. The read operations than return *wrong* values. The *wCF$_{rd}$&wRDF* consists of four FPs: $<0;\ r0:r0/\uparrow/1>$, $<0;\ r1:r1/\downarrow/0>$, $<1;\ r0:r0/\uparrow/1>$, and $<1;r1:r1/\downarrow/0>$.

The 2PF2$_\mathrm{av}$: This type consists of three FFMs, caused by bridges between bit lines of different ports belonging to the same or to adjacent columns [13].

1. *wCF$_{ds}$&wRDF*: A read operation applied to cell c_v flips the cell and the sense amplifier returns an incorrect value *if* a write operation is applied to cell c_a *simultaneously*. A wCF$_{ds}$&wRDF consists of four FPs: $<w0:r0/\uparrow/1>$, $<w0:r1/\downarrow/0>$, $<w1:r0/\uparrow/1>$, and $<w1:r1/\downarrow/0>$.
2. *wCF$_{ds}$&wIRF*: A read operation applied to cell c_v returns an incorrect value *if* a write operation is applied to cell c_a *simultaneously*. It should be noted that the state of cell c_v does not change. A wCF$_{ds}$&wIRF consists of four FPs: $<w0:r0/0/1>$, $<w0:r1/1/0>$, $<w1:r0/0/1>$, and $<w1:r1/1/0>$.
3. *wCF$_{ds}$&wRRF*: A read operation applied to cell c_v returns a random value *if* a write operation is applied to cell c_a *simultaneously*. Note also here that the state of cell c_v does not change. A wCF$_{ds}$&wRRF consists of four FPs: $<w0:r0/0/?>$, $<w0:r1/1/?>$, $<w1:r0/0/?>$, and $<w1:r1/1/?>$.

It should be noted that the above 2PFs are valid for 2P memories which support simultaneous reading and writing of the same location, whereby the read data is discarded. If this is not supported, then the FFM: wRDF&wTF will not be realistic. In addition, the FFM: wCF$_{ds}$&wCF$_{ds}$ will consist only of the FPs sensitized by simultaneous read operations to the same location.

3 Conditions for Detecting 2PFs

This section gives conditions for detecting the 2PFs listed in the previous section; they will be used to derive tests. In order to describe conditions (tests) for 2PFs, the march notation used for SP memories has to be extended in order to be able to specify tests for MP memories.

3.1 Notation for March Tests

The extension will be done as follows:

- A complete march test is delimited by the "$\{\ldots\}$" bracket pair; while a march element is delimited by the "$(\ldots)$" bracket pair. The march elements are separated by semicolons and the operations within a march element are separated by commas.
- The operations applied in parallel to the ports are separated using colons, and the port number to which each of the set of the parallel operations is applied is determined implicitly. For example, the march element $(r0:w1)$ denotes two simultaneous operations: a $r0$ operation applied to the first port (P1), and a $w1$ operation applied to the second port (P2).
- The character "n" denotes *no operation*, while the character "$-$" denotes *any allowed operation*. For example, $(r0:n)$ denotes a $r0$ operation via $P1$, and no operation on $P2$.
- The cell to which the operation is applied can be specified explicitly by subscripting the corresponding operation, e.g., $(r0_{r,c})$ denotes a $r0$ operation applied to the cell in row r and column c.
- $\Updownarrow$: denotes an up addressing ($\Uparrow$), or a down addressing ($\Downarrow$) sequence. $\Uparrow_{r_1=0}^{R-1}\Uparrow_{r_2=r_1+1}^{R-1}$: denotes a nested addressing sequence, whereby r_1 goes from 0 to $R-1$; and for each value of r_1, r_2 goes from r_1+1 to $R-1$.

3.2 Conditions for Detecting 2PFs

The 2PFs are divided into 2PF1s and 2PF2s. The conditions for detecting such faults will be discussed separately; they will be used to derive *functional tests*.

3.2.1 Conditions for Detecting 2PF1s

The 2PF1 fault class consists three FFMs: wDRDF&wDRDF, wRDF&wRDF, and wRDF&wTF; see Fig. 4.

Conditions for Detecting wDRDF&wDRDF. The wDRDF&wDRDF consists of two fault primitives (FPs): $<r0:r0/\uparrow/0>_v$ and $<r1:r1/\downarrow/1>_v$; see Table 3. It has to be considered only for 2P memories which allow for two simultaneous read operations of a same location (i.e., address). It is detectable if the following condition is satisfied:

Condition wDRDF&wDRDF: Any wDRDF&wDRDF fault will be detected by a march test which contains the two march elements of Case A and the two march elements of Case B; these four march elements may be combined into three, two, or a single march element.

- Case A (to detect $<r0:r0/\uparrow/0>_v$):

$$\Updownarrow(\ldots,r0:r0);\Updownarrow(r0:-,\ldots).$$

- Case B (to detect $<r1:r1/\downarrow/1>_v$):

$$\Updownarrow(\ldots, r1 : r1); \Updownarrow(r1 : -, \ldots).$$

The first pair of simultaneous read operations through the two ports in each first march element sensitizes the fault, which will be detected by the single read operation of each second march element. Note that the single read operation can be replaced by simultaneous read operations (i.e., "$-$", which denotes any allowed operation, can be replaced with a read operation).

Conditions for Detecting wRDF&wRDF. The wRDF&wRDF consists of two FPs: $< r0 : r0/\uparrow/1 >_v$ and $< r1 : r1/\downarrow/0 >_v$. It has to be considered only for 2P memories which allow for two simultaneous read operations of a same location. It is detectable if the following condition is satisfied:

Condition wRDF&wRDF: Any wRDF&wRDF fault is detectable by a march test if the test contains the march element of Case A and the march element of Case B; these two march elements can be combined into a single march element.

- Case A (to detect $< r0 : r0/\uparrow/1 >_v$):

$$\Updownarrow(\ldots, r0 : r0, \ldots).$$

- Case B (to detect $< r1 : r1/\downarrow/0 >_v$):

$$\Updownarrow(\ldots, r1 : r1, \ldots).$$

The pair of simultaneous read operations through the two ports in each march element sensitize and detect the fault. Note that any test detecting wDRDF&wDRDF will also detect wRDF&wRDF, since Condition wRDF&wRDF is a subset of Condition wDRDF&wDRDF.

Conditions for Detecting wRDF&wTF. The wRDF&wTF consists of two FPs: $< r0 : w\uparrow/0/- >_v$ and $< r1 : w\downarrow/1/- >_v$. It only applies to 2P memories which allow for simultaneous read and write of the same location, whereby the read data will be discarded. Such faults are detectable if the following condition is satisfied:

Condition wRDF&wTF: Any wRDF&wTF fault will be detected by a march test which contains the pair of march elements of Case A and of Case B. The individual march elements may be combined into one, two, or three march elements.

- Case A (to detect $< r0 : w\uparrow/0/- >_v$):

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

- Case B (to detect $< r1 : w\downarrow/1/- >_v$):

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

The first pair of simultaneous operations through the two ports in each march element sensitize the fault, which will be detected by the second single read operations. Note that the single read operations can be replaced by simultaneous read operations (i.e., "$-$," which denotes any allowed operation, can be replaced with a read operation).

Conditions for Detecting all 2PF1s. The above three conditions can be merged into a single condition; the result, referred to as Condition PF1, is shown below. A test satisfying this condition detects all 2PF1 faults.

Condition 2PF1: Any 2PF1 fault will be detected by a march test which contains both pairs of march elements of Case A (i.e., A.1 and A.2), or both pairs of march elements of Case B (i.e., B.1 and B.2). The two pairs of march elements A.1 and A.2 can be combined into one, two, three, four, or five march elements; while B.1 and B.2 can be combined into one, two, three, four, five, six, or seven elements.

- Case A:

1. To detect $< r0 : r0/\uparrow/0 >_v$, $< r0 : r0/\uparrow/1 >_v$, and $< r1 : w\downarrow/1/- >_v$:

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

2. To detect $< r1 : r1/\downarrow/1 >_v$, $< r1 : r1/\downarrow/0 >_v$, and $< r0 : w\uparrow/0/- >_v$:

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

- Case B:

1. To detect $< r0 : r0/\uparrow/0 >_v$, $< r0 : r0/\uparrow/1 >_v$, and $< r0 : w\uparrow/0/- >_v$:

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

2. To detect $< r1 : r1/\downarrow/1 >_v$, $< r1 : r1/\downarrow/0 >_v$, and $< r1 : w\downarrow/1/- >_v$:

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

It should be noted that the Condition 2PF1 applies to a 2P memory supporting simultaneous read and write of the same location. If this is not the case, then the FFM wRDF&wTF is not realistic; and as a consequence, Condition 2PF1 can be simplified to Condition wDRDF&wDRDF.

3.2.2 Conditions for Detecting 2PF2s

Depending on to which cells (to the a-cell and/or to the v-cell) the two simultaneous operations are applied, the 2PF2 class is divided into three types: $2PF2_a$, $2PF2_v$, and $2PF2_{av}$; see Fig. 3.

Conditions for Detecting $\mathbf{2PF2_a}$s. The $2PF2_a$ consists of one FFM (i.e., wCF_{ds}&wCF_{ds}) with eight FPs: $< w0 : \mathrm{r}d; 0/\uparrow/- >$, $< w0 : \mathrm{r}d; 1/\downarrow/- >$, $< w1 : \mathrm{r}d; 0/\uparrow/- >$, $< w1 : \mathrm{r}d; 1/\downarrow/- >$, $< \mathrm{r}x : \mathrm{r}x; 0/\uparrow/- >$, and $< \mathrm{r}x : \mathrm{r}x; 1/\downarrow/- >$; whereby $x \in \{0, 1\}$ and d is the don't care value. In order to detect any $2PF2_a$, we have:

1. to apply sensitizing operations to a cell c_a; $a \in \{0, 1, 2, \ldots, n-2, n-1\}$ and
2. to detect the fault in cell c_v; $v \neq i$.

The order in which c_a has to be selected is not relevant. Therefore, the $\Updownarrow_{a=0}^{n-1}$ address order can be specified.

Condition 2PF2$_a$: Any 2PF2$_a$ is detectable by a march test which contains both march elements of Case A and both march elements of Case B.

- Case A:

$$\Updownarrow_{a=0}^{n-1} (r0 : r0, \ldots, w1 : \mathrm{r}d, \ldots, r1 : r1, \ldots, w0 : \mathrm{r}d);$$
$$\Updownarrow_{a=0}^{n-1} (r0 : -, \ldots).$$

- Case B:

$$\Updownarrow_{a=0}^{n-1} (r1 : r1, \ldots, w0 : \mathrm{r}d, \ldots, r0 : r0, \ldots, w1 : \mathrm{r}d);$$
$$\Updownarrow_{a=0}^{n-1} (r1 : -, \ldots).$$

Condition 2PF2$_a$ can be explained as follows: The operations in the first march element of Case A will sensitize all 2PF2$_a$ faults when the value of the fault effect is 1, because that march element contains all sensitizing operations. If the address order of a is increasing, then Case A will detect the fault by the "$r0 : r0$" operation of the first march element if the v-cell has a higher address than the a-cell (i.e., $v > a$), and by "$r0$" operation of the second march element if $v < a$. If the address order of a is decreasing, then Case A will detect the fault by the "$r0 : r0$" operation of the first march element if $v < a$, and by "$r0$" operation of the second march element if $v > a$. A similar explanation can be given for Case B, which is required to sensitize and detect 2PF2$_a$ faults when the value of the fault effect is 0.

Note that, in the above condition, simultaneous read and write of the same location is assumed to be supported. If this is not the case, then the 2PF2$_a$ will consist only of FPs sensitized by simultaneous read operations; as a consequence, the operations "wx : rd" in Condition 2PF2$_a$ should be replaced with "w$y : n$," whereby n denotes no operation and $x \in \{0, 1\}$.

Conditions for Detecting 2PF2$_v$s. The 2PF2$_v$ consists of two FFMs: wCF$_{dr}$&wDRDF and wCF$_{rd}$&wRDF, each with four FPs. The wCF$_{dr}$&wDRDF consists of $< 0; r0 : r0/\uparrow/0 >_{a,v}, < 1; r0 : r0/\uparrow/0 >_{a,v}$, $< 0; r1 : r1/\downarrow/1 >_{a,v}$ and $< 1; r1 : r1/\downarrow/1 >_{a,v}$; while the wCF$_{rd}$&wRDF consists of $< 0; r0 : r0/\uparrow/1 >_{a,v}$, $< 1; r0 : r0/\uparrow/1 >_{a,v}$, $< 0; r1 : r1/\downarrow/0 >_{a,v}$, and $< 1; r1 : r1/\downarrow/0 >_{a,v}$. In order to detect the presence of such faults in a cell c_v, the following condition has to be satisfied:

Condition 2PF2$_v$: Any wCF$_{dr}$&wDRDF and wCF$_{rd}$&wRDF is detectable by a march test if the test selects all pairs of cells (c_a, c_v) whereby $a \in \{0, 1, \ldots, v-1, v+1, \ldots, n-2, n-1\}$ and each pair undergoes the four states 00, 01, 10, and 11. In addition, in each state, two simultaneous read operations, followed by at least a single read operation, have to be applied to the v-cell.

Condition 2PF2$_v$ can be explained as follows: When the pair of cells (c_a, c_v) is in state 00, then the fault $< 0; r0 : r0/\uparrow/1 >_{a,v}$ will be sensitized and detected by simultaneous read operations applied to the v-cell; while the fault $< 0; r0 : r0/\uparrow/0 >_{a,v}$ will be sensitized. The extra (single) read operation will detect the last fault. A similar explanation can be given for the other three states.

Conditions for Detecting 2PF2$_{\mathbf{av}}$s. The 2PF2$_{av}$ consists of three FFMs: wCF$_{ds}$&wRDF, wCF$_{ds}$&wIRF, and wCF$_{ds}$&wRRF; each with four FPs. The wCF$_{ds}$&wRDF consists of $< w1 : r1/\downarrow/0 >_{a,v}$, $< w1 : r0/\uparrow/1 >_{a,v}$, $< w0 : r1/\downarrow/0 >_{a,v}$ and $< w0 : r0/\uparrow/1 >_{a,v}$. The wCF$_{ds}$&wIRF consists of $< w1 : r1/1/0 >_{a,v}$, $< w1 : r0/0/1 >_{a,v}$, $< w0 : r1/1/0 >_{a,v}$ and $< w0 : r0/0/1 >_{a,v}$; while wCF$_{ds}$&wRRF consists of $< w1 : r1/1/? >_{a,v}, < w1 : r0/0/? >_{a,v}$, $< w0 : r1/1/? >_{a,v}$, and $< w0 : r0/0/? >_{a,v}$. In order to detect the presence of such faults in cell c_v, we have to:

1. select all pairs (c_a, c_v), whereby
$$a \in \{0, 1, \ldots, v-1, v+1, \ldots, n-2, n-1\},$$
2. apply sensitizing operations to the two cells, and
3. read the cell c_v.

The order in which c_v will be selected is not important, the only requirement is that v has take on all values from the set $\{0, 1, 2, \ldots, n-2, n-1\}$. The select order can be given as follows: $\Updownarrow_{v=0}^{n-1}$. In addition, the order in which c_a will be selected is also not important; the only requirement is that a has to take on all values from the set $\{0, 1, \ldots, v-1, v+1, \ldots, n-2, n-1\}$. Therefore, the select order can be given as follows: $\Updownarrow_{a=0}^{v-1}$ and $\Updownarrow_{a=v+1}^{n-1}$.

In the above, it is assumed that the a-cell as well as the v-cell can be any cell of the memory cell array. However, this is not the case in real designs. The inductive fault analysis and SPICE simulation [13] show that the 2PF2$_{av}$ can only be caused between adjacent cells in the same row or column; see Fig. 5. That means that if the a-cell is $c_a = c_{r,c}$ (i.e., a cell in row r and column c), then the v-cell has to be $c_{r\pm1,c}$ or $c_{r,c\pm1}$; the distance between a-cell and v-cell is just 1.

In addition, it has also been shown that a defect between cells (say c$_1$ and c$_2$) is symmetrical with respect to the two cells; for example, if the application of the operation "$w0_{c1} : r1_{c2}$" (i.e., $w1$ applied to c$_1$ and $r1$ applied to c$_2$) causes a certain fault in c$_2$, then the same fault will be caused by applying the operation "$r1_{c1} : w0_{c2}$." The only difference is that, in the first case, c$_2$ is the v-cell while in the second case c$_1$ is the v-cell. Therefore, for each a-cell $c_{r,c}$, one can restrict the v-cell space to:

a. $c_{r+1,c}$ and $c_{r,c+1}$,
b. $c_{r-1,c}$ and $c_{r,c+1}$,
c. $c_{r+1,c}$ and $c_{r,c-1}$, or
d. $c_{r-1,c}$ and $c_{r,c-1}$.

Note that, e.g., if for the a-cell $= c_{r,c}$ the two v-cells $c_{r+1,c}$ and $c_{r,c+1}$ are considered, then one does not need to consider $c_{r,c}$ as a v-cell when a-cell $= c_{r+1,c}$ (or $c_{r,c+1}$). This

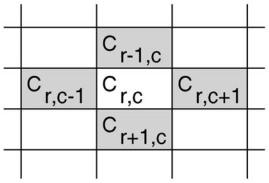


Fig. 5. The v-cell space for $\text{a-cell} = c_{r,c}$.

is because a defect between $c_{r,c}$ and $c_{r+1,c}$ is symmetrical with respects to the two cells. When $\text{a-cell} = c_{r+1,c}$, then the two cells $c_{r+2,c}$ and $c_{r+1,c+1}$ have to be considered as v-cells; see Fig. 5. This reduction in a-cell and v-cell spaces has a significantly impact on the condition to detect the $2PF2_{av}$ faults and, therefore, on the test. It reduces the time complexity from $\Theta(n^2)$ to $\Theta(n)$. However, it requires the test to use topological addressing, rather than logical addressing [16].

Condition 2PF2$_{\mathbf{av}}$: Any wCF_{ds}&wRDF and wCF_{ds}&wIRF is detectable by a march test if the test contains one of the four pairs of march elements of Case A (e.g., the pair of march elements of Case A.1 or of Case A.2, A.3, or A.4), of Case B, of Case C, and of Case D. These four pairs of march elements can be combined into one, two, three, four, five, six, or seven march elements. In addition, a march test satisfying this condition can also probabilistically detect wCF_{ds}&wRRF. That means that the detection of this fault cannot be guaranteed due to the fact that the read operation produces a random value. Note that a nested addressing sequence is used, e.g., $\Uparrow_{c=0}^{C-1} \Uparrow_{r=0}^{R-1}$, whereby R and C denote the number of rows and columns in the memory cell array, respectively. The column c goes from 0 to $C-1$ ($\Uparrow_{c=0}^{C-1}$) or from $C-1$ to 0 ($\Downarrow_{0}^{c=C-1}$); and for each value of c, row r goes from 0 to $R-1$ ($\Uparrow_{r=0}^{R-1}$) or from $R-1$ to 0 ($\Downarrow_{0}^{r=R-1}$).

- Case A (to detect $< w1 : r1/\downarrow/0 >_{a,v}$ and $< w1 : r1/1/0 >_{a,v}$)

A.1.

$$\Uparrow_{c=0}^{C-1} (\Uparrow_{r=0}^{R-1} (\ldots, w1_{r,c} : r1_{r+1,c}, \ldots));$$
$$\Uparrow_{c=0}^{C-1} (\Uparrow_{r=0}^{R-1} (\ldots, w1_{r,c} : r1_{r,c+1}, \ldots)).$$

A.2.

$$\Uparrow_{c=0}^{C-1} (\Downarrow_{0}^{r=R-1} (\ldots, w1_{r,c} : r1_{r-1,c}, \ldots));$$
$$\Uparrow_{c=0}^{C-1} (\Downarrow_{0}^{r=R-1} (\ldots, w1_{r,c} : r1_{r,c+1}, \ldots)).$$

A.3.

$$\Downarrow_{0}^{c=C-1} (\Uparrow_{r=0}^{R-1} (\ldots, w1_{r,c} : r1_{r+1,c}, \ldots));$$
$$\Downarrow_{0}^{c=C-1} (\Uparrow_{r=0}^{R-1} (\ldots, w1_{r,c} : r1_{r,c-1}, \ldots)).$$

A.4.

$$\Downarrow_{0}^{c=C-1} (\Downarrow_{0}^{r=R-1} (\ldots, w1_{r,c} : r1_{r-1,c}, \ldots));$$
$$\Downarrow_{0}^{c=C-1} (\Downarrow_{0}^{r=R-1} (\ldots, w1_{r,c} : r1_{r,c-1}, \ldots)).$$

- Case B: (to detect $< w1 : r0/\uparrow/1 >_{a,v}$ and $< w1 : r0/0/1 >_{a,v}$)

B.1.

$$\Uparrow_{c=0}^{C-1} (\Uparrow_{r=0}^{R-1} (\ldots, w1_{r,c} : r0_{r+1,c}, \ldots));$$
$$\Uparrow_{c=0}^{C-1} (\Uparrow_{r=0}^{R-1} (\ldots, w1_{r,c} : r0_{r,c+1}, \ldots)).$$

B.2.

$$\Uparrow_{c=0}^{C-1} (\Downarrow_{0}^{r=R-1} (\ldots, w1_{r,c} : r0_{r-1,c}, \ldots));$$
$$\Uparrow_{c=0}^{C-1} (\Downarrow_{0}^{r=R-1} (\ldots, w1_{r,c} : r0_{r,c+1}, \ldots)).$$

B.3.

$$\Downarrow_{0}^{c=C-1} (\Uparrow_{r=0}^{R-1} (\ldots, w1_{r,c} : r0_{r+1,c}, \ldots));$$
$$\Downarrow_{0}^{c=C-1} (\Uparrow_{r=0}^{R-1} (\ldots, w1_{r,c} : r0_{r,c-1}, \ldots)).$$

B.4.

$$\Downarrow_{0}^{c=C-1} (\Downarrow_{0}^{r=R-1} (\ldots, w1_{r,c} : r0_{r-1,c}, \ldots));$$
$$\Downarrow_{0}^{c=C-1} (\Downarrow_{0}^{r=R-1} (\ldots, w1_{r,c} : r0_{r,c-1}, \ldots)).$$

- Case C: (to detect $< w0 : r1/\downarrow/0 >_{a,v}$ and $< w0 : r1/1/0 >_{a,v}$)

C.1.

$$\Uparrow_{c=0}^{C-1} (\Uparrow_{r=0}^{R-1} (\ldots, w0_{r,c} : r1_{r+1,c}, \ldots));$$
$$\Uparrow_{c=0}^{C-1} (\Uparrow_{r=0}^{R-1} (\ldots, w0_{r,c} : r1_{r,c+1}, \ldots)).$$

C.2.

$$\Uparrow_{c=0}^{C-1} (\Downarrow_{0}^{r=R-1} (\ldots, w0_{r,c} : r1_{r-1,c}, \ldots));$$
$$\Uparrow_{c=0}^{C-1} (\Downarrow_{0}^{r=R-1} (\ldots, w0_{r,c} : r1_{r,c+1}, \ldots)).$$

C.3.

$$\Downarrow_{0}^{c=C-1} (\Uparrow_{r=0}^{R-1} (\ldots, w0_{r,c} : r1_{r+1,c}, \ldots));$$
$$\Downarrow_{0}^{c=C-1} (\Uparrow_{r=0}^{R-1} (\ldots, w0_{r,c} : r1_{r,c-1}, \ldots)).$$

C.4.

$$\Downarrow_{0}^{c=C-1} (\Downarrow_{0}^{r=R-1} (\ldots, w0_{r,c} : r1_{r-1,c}, \ldots));$$
$$\Downarrow_{0}^{c=C-1} (\Downarrow_{0}^{r=R-1} (\ldots, w0_{r,c} : r1_{r,c-1}, \ldots)).$$

- Case D: (to detect $< w0 : r0/\uparrow/1 >_{a,v}$) and $< w0 : r0/0/1 >_{a,v}$

D.1.

$$\Uparrow_{c=0}^{C-1} (\Uparrow_{r=0}^{R-1} (\ldots, w0_{r,c} : r0_{r+1,c}, \ldots));$$
$$\Uparrow_{c=0}^{C-1} (\Uparrow_{r=0}^{R-1} (\ldots, w0_{r,c} : r0_{r,c+1}, \ldots)).$$

D.2.

$$\Uparrow_{c=0}^{C-1} (\Downarrow_{0}^{r=R-1} (\ldots, w0_{r,c} : r0_{r-1,c}, \ldots));$$
$$\Uparrow_{c=0}^{C-1} (\Downarrow_{0}^{r=R-1} (\ldots, w0_{r,c} : r0_{r,c+1}, \ldots)).$$

D.3.

$$\Downarrow_{0}^{c=C-1} (\Uparrow_{r=0}^{R-1} (\ldots, w0_{r,c} : r0_{r+1,c}, \ldots));$$
$$\Downarrow_{0}^{c=C-1} (\Uparrow_{r=0}^{R-1} (\ldots, w0_{r,c} : r0_{r,c-1}, \ldots)).$$

D.4.

$$\Downarrow_{0}^{c=C-1} (\Downarrow_{0}^{r=R-1} (\ldots, w0_{r,c} : r0_{r-1,c}, \ldots));$$
$$\Downarrow_{0}^{c=C-1} (\Downarrow_{0}^{r=R-1} (\ldots, w0_{r,c} : r0_{r,c-1}, \ldots)).$$

Condition $2\text{PF}2_{av}$ can be explained as follows: The operation "$w1_{r,c} : r1_{r+1,c}$" in the first pair of march elements of Case A.1 will sensitize and detect the faults $< w1 : r1/\downarrow/0 >_{a,v}$, $< w1 : r1/1/0 >_{a,v}$, and may detect $< w1 : r1/1/? >_{a,v}$, whereby the v-cell (i.e., $c_{r+1,c}$) and the a-cell ($c_{r,c}$) are adjacent in the same column; while the operation "$w1_{r,c} : r1_{r,c+1}$" will detect the same faults when the v-cell and the a-cell ($c_{r,c+1}$) are adjacent in the same row. It should be noted that, in case $r = R$ (or $c = C$), then $r + 1$ should be replaced with $(r+1)mod^1 R$ (and $c + 1$ with $(c+1) mod\, C$). The other three pairs of march elements of Case A detect the same faults as the first pair; however, they use different addressing orders. For example, Case A.2 considers the a-cell $c_{r,c}$ and the two v-cells $c_{r-1,c}$ and $c_{r,c+1}$, etc. A similar explanation applies to Case B, Case C, and Case D.

4 TESTS FOR TWO-PORT FAULTS

The FFMs for 2P memory cell array faults are divided into 1PFs and 2PFs. The 1PFs are the conventional faults that can occur in SP memories; while the 2PFs are faults that are based on simultaneous operations applied to the 2P memory. Therefore, the test procedure can be divided into two parts:

1. Test(s) to detect 1PFs.
2. Test(s) to detect 2PFs.

Tests for 1PFs are described in literature [14], [15], [16], [17]; while tests for 2PFs have to be derived. Below, tests for detecting 2PFs will be introduced. First, tests for 2PF1s will be presented and thereafter for 2PF2s.

4.1 Tests for the 2PF1s

The test shown in Fig. 6, referred as *March 2PF1*, detects all 2PF1 faults. March 2PF1 consists of three march elements: M_0, M_1, and M_2; it has a test length of $7n$, whereby n is the size of the memory cell array. Note that the single read operations in M_1 and M_2 can be replaced with simultaneous read operations, e.g., "$r1 : -$" can be replaced with "$r1 : r1$."

March 2PF1 detects all 2PF1 faults since it satisfies Condition 2PF1 of Section 3.2.1. The second march element of the test (i.e., M_1) contains the pairs of march elements of Case A.2, while M_2 contains the pairs of march elements A.1. The three march elements of the test can be merged into one or two elements without impacting the fault coverage.

It should be noted that simultaneous read and write operations of the same location (with read data discarded) is assumed to be allowed. If this is not supported, then the FFM wRDF&wTF is not realistic; and as a consequence, March 2PF1 can be simplified. The result is shown in Fig. 7, and referred as *March 2PF1-*; it has a test length of $6n$. Note that the four march elements of the test can be merged into one, two, or three march elements.

4.2 Tests for the 2PF2s

The 2PF2s are divided, depending on the cells to which the simultaneous operations are applied, into three types: $2\text{PF}2_a$, $2\text{PF}2_v$, and $2\text{PF}2_{av}$; see Fig. 3. In the following, tests for each type will be introduced.

4.2.1 Tests for the $\mathbf{2PF2_a}$ Faults

March $2PF2_a$, shown in Fig. 8, detects all $2\text{PF}2_\text{a}$ faults since it satisfies Condition $2\text{PF}2_\text{a}$ of Section 3.2: M_1 and M_2 contain the two march elements of Case A, while M_3 and M_4 contain the two march elements of Case B. Note that the operations "$r0 : -$" and "$r1 : -$" in the test can be replaced with "$r0 : r0$," respectively, with "$r1 : r1$." March $2\text{PF}2_a$ has a test length of $12n$, and can also detect the 2PF1s: wRDF&wRDF and wRDF&wTF; see also Table 5.

March $2\text{PF}2_a$ can be further optimized to a $10n$ test without impacting the fault coverage. The result is shown in Fig. 9, and referred as *March* $2PF2_a-$; note that the operation "$r0 : -$" in M_5 can be replaced with "$r0 : r0$." Although the test of Fig. 9 does not explicitly contain the two pairs of march elements of Condition $2\text{PF}2_a$, it detects all $2\text{PF}2_a$ faults; the sensitizing operations for $2\text{PF}2_a$ faults are distributed over M_1, M_2, M_3, M_4, and M_5.

The $2\text{PF}2_a$ faults consists of one FFM: wCF_{ds}&wCF_{ds} with eight FPs:

$$< w0 : \text{r}d; 0/\uparrow/- >, < w0 : \text{r}d; 1/\downarrow/- >,$$
$$< w1 : \text{r}d; 0/\uparrow/- >, < w1 : \text{r}d; 1/\downarrow/- >,$$
$$< \text{r}x : \text{r}x; 0/\uparrow/- >$$

and $< \text{r}x : \text{r}x; 1/\downarrow/- >$; whereby $x \in \{0, 1\}$ and d is the don't care value.

- The $< r0 : r0; 0/\uparrow/- >_{a,v}$ and the $< w1 : \text{r}d; 0/\uparrow/- >_{a,v}$ will be sensitized and detected by M_1 of Fig. 9 if the v-cell has a higher address than the a-cell, i.e., $v > a$. If $v < a$, then these faults will be sensitized and detected by M_3.
- The $< r1 : r1; 1/\downarrow/- >_{a,v}$ and $< w0 : \text{r}d; 1/\downarrow/- >_{a,v}$ will be sensitized and detected by M_2 if $v > a$. If $v < a$, then these faults will be sensitized and detected by M_4.
- The $< r0 : r0; 1/\downarrow/- >_{a,v}$ and the $< w1 : \text{r}d; 1/\downarrow/- >_{a,v}$ will be sensitized by M_1 and detected by M_2

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

Fig. 6. March 2PF1.

if $v < a$; while the same faults will be sensitized by M_3 and detected by M_4 if $v > a$.

- The $< r1:r1;0/\uparrow/- >_{a,v}$ and the $< w0:\mathrm{r}d;0/\uparrow/- >_{a,v}$ will be sensitized by M_2 and detected by M_3 if $v < a$; while the same faults will be sensitized by M_4 and detected by M_5 if $v > a$.

Again, if simultaneous read and write is not supported, then the operations "wx : rd" in March $2\mathrm{PF}2_a$ (and March $2\mathrm{PF}2_a -$) should be replaced with "wx : n." In that case, the $2\mathrm{PF}2_a$ will consist only of FPs based on two simultaneous read operations.

4.2.2 Tests for the $\mathbf{2PF2}_v$ Faults

The test detecting all $2\mathrm{PF}2_v$ faults is shown in Fig. 10, and is referred as *March* $2PF2_v$; this test satisfies Condition $2\mathrm{PF}2_v$ of Section 3.2.2. M_0 initializes all memory cells to 0; which means that the state of all pairs (c_a, c_v) is 00. In the second march element, the v-cell is first read via the two ports simultaneously while the state of all pairs (c_a, c_v) is 00; therefore, the fault $< 0;r0:r0/\uparrow/1 >_{a,v}$ will be detected, while the fault $< 0;r0:r0/\uparrow/0 >_{a,v}$ will be sensitized (see Table 3). The latter will be detected by the next operation within the same march element. Thereafter, the v-cell will be written with 1, and then will be read via the two ports simultaneously. This means that simultaneous read operations are applied to v-cell while the state of all pairs (c_a, c_v) is 01; therefore, the fault $< 0;r1:r1/\downarrow/0 >_{a,v}$ will be detected, while the fault $< 0;r1:r1/\downarrow/1 >_{a,v}$ will be sensitized. The latter fault will be detected by the next read operation within the same march element. Finally, the v-cell is written with 0, such that all pairs (c_a, c_v) again enter state 00. A similar explanation can be given for march elements M_2 and M_3. Note that the operations "$r0:-$" and "$r1:-$" in M_1 and M_3 can be replaced with "$r0:r0$," respectively, "$r1:r1$;" and that March $2\mathrm{PF}2_v$ has a test length of $14n$. Note also that March $2\mathrm{PF}2_v$ also detects the 2PF1 faults: wDRDF&wDRDF and wRDF&wRDF (see also Fig. 6 and Table 3).

March $2\mathrm{PF}2_v$ can also be optimized without impacting the fault coverage of the $2\mathrm{PF}2_v$ faults. The result is shown in Fig. 11, and referred as *March* $2PF2_v -$. It consists of five march elements and has a test length of $13n$. Note that the addressing sequence of the optimized version is *relevant*; it is important for the fault coverage as it will be shown below. Note also that the operations "$r0:-$" and "$r1:-$" in the test can be replaced with "$r0:r0$," respectively, "$r1:r1$."

$$\{\ \underset{M_0}{\Updownarrow(w0:n)}\ :\ \underset{M_1}{\Updownarrow(r0:r0,r0:-)}\ :\ \underset{M_2}{\Updownarrow(w1:-)}\ :\ \underset{M_3}{\Updownarrow(r1:r1,r1:-)}\ \}$$

Fig. 7. March 2PF1-.

$$\{\ \underset{M_0}{\Updownarrow(w0:n)}\ ;\ \underset{M_1}{\Updownarrow(r0:r0,w1:r0,r1:r1,w0:r1)}\ ;\ \underset{M_2}{\Updownarrow(r0:-,w1:-)}\ ;\ \underset{M_3}{\Updownarrow(r1:r1,w0:r1,r0:r0,w1:r0)}\ ;\ \underset{M_4}{\Updownarrow(r1:-)}\ \}$$

Fig. 8. March $2\mathrm{PF}2_a$.

March $2\mathrm{PF}2_v -$ detects all $2\mathrm{PF}2_v$ faults since Condition $2\mathrm{PF}2_v$ of Section 3.2.2 is satisfied. Table 4 shows the operations performed on two cells c_i and c_j by the march elements of Fig. 11. The table contains a column "state," which identifies the state $S_{i,j}$ of the two cells (c_i, c_j) *before* the operation is performed; and a column "State $S_{i,j}$," which identifies the state *after* the operation. The table shows that all states of (c_i, c_j) (i.e., 00, 01, 11, 10) are generated and, in each state, two simultaneous read operations followed by (at least) a single read operation are applied to cell c_i and c_j.

4.2.3 Tests for the $\mathbf{2PF2}_{av}$ Faults

The test shown in Fig. 12, referred to as *March* $2PF_{av}$, detects all wCF_{ds}&wRDF faults, all wCF_{ds}&wIRF faults, and also, probabilistically detects wCF_{ds}&wRRF faults since it satisfies Condition $2\mathrm{PF}2_{av}$ of Section 3.2.2: M_1 contains the first march element of Case B.1 and of Case D.1; M_2 contains the second march element of Case B.1 and of Case D.1. In addition, M_4 contains the first march element of Case C.1 and of Case A.1; while M_5 contains the second march elements of Case C.1 and of Case A.1. March $2\mathrm{PF}_{av}$ has a test length of $10n$.

Below, an optimal version of March $2\mathrm{PF}2_{av}$ will be given. The test, referred to as *March* $2PF2_{av} -$, is shown in Fig. 13; it has a test length of $9n$ and consists of nine march elements, each with only one operation. It can be clearly seen that the test, similar to March $2\mathrm{PF}2_{av}$, covers the $2\mathrm{PF}2_{av}$ faults since it satisfies Condition $2\mathrm{PF}2_{av}$ of Section 3.2.2: M_1, M_2, M_3, and M_4 contain the first march elements of Case B.1, Case A.1, Case C.1, and Case D.1, respectively; while M_5, M_6, M_7, and M_8 contain the second march elements of Case B.1, Case A.1, Case C.1, and Case D.1, respectively.

Another optimal version of March $2\mathrm{PF}2_{av}$ is shown in Fig. 14. It has also a test length of $9n$, but it consists only of three march elements. The test satisfies Condition $2\mathrm{PF}2_{av}$ of Section 3.2.2 as follows: M_2 contains the first march elements of Case B.1, Case A.1, Case C.1, and Case D.1; while M_3 contains the second march elements of Case B.1, Case A.1, Case C.1, and Case D.1. Note that the read and the write operations are interchanged in some cases, e.g., "$w1:r0$" changed into "$r0:w1$." This has no negative

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

Fig. 9. March $2\mathrm{PF}2_a -$.

$$\{ \underbrace{\Updownarrow (w0:n)}_{M_0} ; \underbrace{\Updownarrow (r0:r0, r0:-, w1:-, r1:r1, r1:-, w0:-)}_{M_1} ; \underbrace{\Updownarrow (w1:-)}_{M_2} ; \underbrace{\Updownarrow (r1:r1, r1:-, w0:-, r0:r0, r0:-, w1:-)}_{M_3} \}$$

Fig. 10. March $2\text{PF}2_v$.

impact on the fault coverage of the $2\text{PF}2_{av}$ faults since, if a certain defect between adjacent cells (say c_1 and c_2) causes a fault sensitized by "$w1_{c1} : r0_{c2}$" (i.e., $w1$ applied to c_1 and $r1$ applied to c_2), then the same fault will also be sensitized with "$r0_{c1} : w1_{c2}$;" the only difference is that, in the first case, c_2 is the victim cell (i.e, the cell where the fault effect appears), while, in the second case c_1 is the victim cell [13].

4.3 Classification of 2PF Tests

The proposed march tests for 2P memories can be classified, based on the type of addressing they use, into two classes:

- *Single-addressing tests*: These are tests which access one cell at a time (i.e., both ports use the same address). They consist of March 2PF1, March $2\text{PF}2_a$, and March $2\text{PF}2_v$.
- *Double-addressing tests*: These are tests which access two different locations at a time. They consist only of March $2\text{PF}2_{av}$. From here on, this test will be referred to as *March d2PF*.

The three single-addressing tests can be merged into a single march test; the result is shown in Fig. 15 and is referred as *March s2PF* (*s* for single addressing). The test satisfies Condition 2PF1 (of Section 3.2.1) required to detect 2PF1s; it also satisfies Condition $2\text{PF}2_a$ and Condition $2\text{PF}2_v$ of Section 3.2.2.

- Condition 2PF1: satisfied by M_1 and M_2, as well as by M_4 and M_5.
- Condition $2\text{PF}2_a$: satisfied by M_1, M_2, M_4, and M_5. M_1 and M_2 contain the two march elements of Case A; while M_4 and M_5 contain the two march elements of Case B.
- Condition $2\text{PF}2_v$: satisfied by M_1 and M_4. Note that M_1 and M_4 are extended versions of M_1 and M_3 of March $2\text{PF}2_v$; see Fig. 10.

It has to be clear from Fig. 15 that March s2PF has a test length of $16n$; while the test lengths of March 2PF1, March $2\text{PF}2_a$, and March $2\text{PF}2_v$ are $7n$, $12n$, and $14n$, respectively. Therefore, in order to detect 2PF1, $2\text{PF}2_a$, and $2\text{PF}2_v$ faults, one can use March s2PF instead of testing for these faults separately. This will reduce the test time with 51.15%, i.e., from $7n + 12n + 14n = 33n$ to $16n$.

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

Fig. 11. March $2\text{PF}2_v-$.

TABLE 4
State Table for Detecting $2\text{PF}2_v$ Faults

Step	March element	State	Operation	State $S_{i,j}$
1	M_0	$--$	'$w0:n$' to c_i	$0-$
2		$0-$	'$w0:n$' to c_j	00
3	M_1	00	'$r0:r0$' to c_i	00
4		00	'$r0:-$' to c_i	00
5		00	'$w1:-$' to c_i	10
6		10	'$r0:r0$' to c_j	10
7		10	'$r0:-$' to c_j	10
8		10	'$w1:-$' to c_j	11
9	M_2	11	'$r1:r1$' to c_i	11
10		11	'$r1:-$' to c_i	11
11		11	'$w0:-$' to c_i	01
12		01	'$r1:r1$' to c_j	01
13		01	'$r1:-$' to c_j	01
14		01	'$w0:-$' to c_j	00
15	M_3	00	'$r0:r0$' to c_j	00
16		00	'$r0:-$' to c_j	00
17		00	'$w1:-$' to c_j	01
18		01	'$r0:r0$' to c_i	01
19		01	'$r0:-$' to c_i	01
20		01	'$w1:-$' to c_i	11
21	M_4	11	'$r1:r1$' to c_j	11
22		11	'$r1:-$' to c_j	11
23		11	'$w0:-$' to c_j	10
24		10	'$r1:r1$' to c_i	10
25		10	'$r1:-$' to c_i	10
26		10	'$w0:-$' to c_i	00

March s2PF can be further optimized without impacting the fault coverage by splitting M_1 and M_4 each into two march elements. Fig. 16 shows the result, referred to as *March s2PF-*. It consists of six march elements and has a test length of $14n$, i.e., $2n$ less than March s2PF. That means that if this version is used to detect 2PF1, $2\text{PF}2_a$, and $2\text{PF}2_v$ faults, then the test time reduction will be 57.57% (i.e., from $33n$ to $14n$). Note that the addressing sequence of the optimized version is *relevant*; it is important for the fault coverage.

March s2PF- detects all 2PF1s since it satisfies Condition 2PF1 of Section 3.2.1: M_1 and M_2 (also M_3 and M_4) contain the pair B.1, while M_2 and M_3 (also M_4 and M_5) contain the pair B.2. In addition, it detects all 2PF_a faults. Note that the march elements of March

TABLE 5
Summary of the 2P Tests

Test	T.L.	Fault coverage
March 2PF1	$7n$	All 2PF1s
March $2\text{PF}2_a$	$12n$	All $2\text{PF}2_a$s wRDF&wRDF, wRDF&wTF
March $2\text{PF}2_a$-	$10n$	All $2\text{PF}2_a$s wRDF&wRDF, wRDF&wTF
March $2\text{PF}2_v$	$14n$	All $2\text{PF}2_v$s wDRDF&wDRDF, wRDF&wRDF
March $2\text{PF}2_v$-	$13n$	All $2\text{PF}2_v$s wDRDF&wDRDF, wRDF&wRDF
March $2\text{PF}2_{av}$	$10n$	All $2\text{PF}2_{av}$s
March $2\text{PF}2_{av}$-	$9n$	All $2\text{PF}2_{av}$s
March s2PF	$16n$	All 2PF1s, all $2\text{PF}2_a$s, all $2\text{PF}2_v$s
March s2PF-	$14n$	All 2PF1s, all $2\text{PF}2_a$s, all $2\text{PF}2_v$s

$$\{\ \underset{M_0}{\Updownarrow(w0:n)}\ ;\ \underset{M_1}{\Uparrow_{c=0}^{C-1}(\Uparrow_{r=0}^{R-1}(w1_{r,c}:r0_{r-1,c},w0_{r,c}:r0_{r+1,c}))}\ :\ \underset{M_2}{\Uparrow_{c=0}^{C-1}(\Uparrow_{r=0}^{R-1}(w1_{r,c}:r0_{r,c+1},w0_{r,c}:r0_{r,c-1}))}\ ;$$
$$\underset{M_3}{\Updownarrow(w1:\ \ )}\ ;\ \underset{M_4}{\Uparrow_{c=0}^{C-1}(\Uparrow_{r=0}^{R-1}(w0_{r,c}:r1_{r-1,c},w1_{r,c}:r1_{r+1,c}))}\ :\ \underset{M_5}{\Uparrow_{c=0}^{C-1}(\Uparrow_{r=0}^{R-1}(w0_{r,c}:r1_{r,c+1},w1_{r,c}:r1_{r,c+1}))}\ \}$$

Fig. 12. March $2\mathrm{PF}2_{a,v}$.

$$\{\ \underset{M_0}{\Updownarrow(w0:n)}\ ;\ \underset{M_1}{\Uparrow_{c=0}^{C-1}(\Uparrow_{r=0}^{R-1}(w1_{r,c}:r0_{r+1,c}))}\ ;\ \underset{M_2}{\Uparrow_{c=0}^{C-1}(\Uparrow_{r=0}^{R-1}(w1_{r,c}:r1_{r+1,c}))}\ ;$$
$$\underset{M_3}{\Uparrow_{c=0}^{C-1}(\Uparrow_{r=0}^{R-1}(w0_{r,c}:r1_{r+1,c}))}\ ;\ \underset{M_4}{\Uparrow_{c=0}^{C-1}(\Uparrow_{r=0}^{R-1}(w0_{r,c}:r0_{r+1,c}))}\ ;$$
$$\underset{M_5}{\Uparrow_{c=0}^{C-1}(\Uparrow_{r=0}^{R-1}(w1_{r,c}:r0_{r,c+1}))}\ ;\ \underset{M_6}{\Uparrow_{c=0}^{C-1}(\Uparrow_{r=0}^{R-1}(w1_{r,c}:r1_{r,c+1}))}\ ;$$
$$\underset{M_7}{\Uparrow_{c=0}^{C-1}(\Uparrow_{r=0}^{R-1}(w0_{r,c}:r1_{r,c+1}))}\ ;\ \underset{M_8}{\Uparrow_{c=0}^{C-1}(\Uparrow_{r=0}^{R-1}(w0_{r,c}:r0_{r,c+1}))}\ \}$$

Fig. 13. March $2\mathrm{PF}2_{a,v}-$ (optimal version 1).

$$\{\ \underset{M_0}{\Updownarrow(w0:n)}\ ;\ \underset{M_1}{\Uparrow_{c=0}^{C-1}(\Uparrow_{r=0}^{R-1}(w1_{r,c}:r0_{r+1,c},r1_{r,c}:w1_{r-1,c},w0_{r,c}:r1_{r-1,c},r0_{r,c}:w0_{r+1,c}))}\ ;$$
$$\underset{M_2}{\Uparrow_{c=0}^{C-1}(\Uparrow_{r=0}^{R-1}(w1_{r,c}:r0_{r,c-1},r1_{r,c}:w1_{r,c-1},w0_{r,c}:r1_{r,c+1},r0_{r,c}:w0_{r,c+1}))}\ \}$$

Fig. 14. March $2\mathrm{PF}2_{av}-$ (optimal version 2).

s2PF2- are the same as those of March $2\mathrm{PF}2_a-$ (see Fig. 9); except M_1, M_2, M_3, and M_4 are extended with read operations (that do not impact the fault coverage). Moreover, all $2\mathrm{PF}2_v$ faults will be detected since M_1 through M_4 of March s2PF- are the same as those of March $2\mathrm{PF}2_v$ (see Fig. 11), with the only difference that "$w0:-$" in M_2 and M_4 is replaced with "$w0:r1$", and "$w1:-$" in M_1 and M_3 with "$w1:r0$;" that does not impact the fault coverage.

It should be noted that for March s2PF and its optimal version, simultaneous read and write of the same location has to be allowed. If this is not the case, then all operations "$\mathrm{w}x:\mathrm{r}y$" in Fig. 15 and Fig. 16 should be replaced with "$\mathrm{w}x:n;$" whereby $x,y\in\{0,1\}$.

4.4 Summary of 2P Tests

Table 5 summarizes the tests introduced in this section. It shows their required number of operations (i.e., test length (T.L.)) including the initialization, together with their fault coverage; see also Fig. 4.

Based on the above classification, one can conclude that the test for 2PFs in 2P memories can be done using two *linear* march tests: March d2PF, i.e., or March $2\mathrm{PF}2_{av}$ (or its optimized version) with a test length of $10n$ $(9n)$, and March s2PF (or its optimized version) with a test length of $16n$ (14n).

5 Test Strategy

The realistic FFMs for 2P memories have been derived and divided into 1PFs and 2PFs. Tests for 1PFs are know in the literature like in [14], [15], [16], [17]; while tests for 2PFs have been devolved in the previous section. Now, the test strategy can be established. Fig. 17 shows the test strategy for 2P memory (with two read write ports). First, the 1PFs are tested by applying SP tests (like March C- [20]). Note that it is necessary to apply the tests through each port separately. This is because the 1PFs can be *cell faults* or *port faults*; whereby, a cell fault is a fault caused by a defect within a memory cell (e.g., an open at the pull down of the cell), while a port fault is a fault caused by a defect related to a certain port (e.g., an open in the bit line of port P_a). The cell faults can be thus tested just via one port; however, port faults requires the test to be applied via each port separately.

After testing 1PFs, the dies may and may not pass the applied SP tests. If the die fail SP test(s), then it does not make sense to test it for 2PFs. However, if the die pass SP test(s), the application of the 2P tests is required. It has been shown in the previous section that all 2PFs can be tested by March s2PF- and March d2PF-. Therefore, one can only use these two tests to detect the 2PFs for the dies passing the SP test(s).

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

Fig. 15. March s2PF for 2PF1s, $2\mathrm{PF}2_a$s, and $2\mathrm{PF}2_v$s.

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

Fig. 16. March s2PF-.

```
Detection of 1PFs:
  Apply single port test(s) via port $P_a$;
  Apply single port test(s) via port $P_b$;
Detection of 2PFs:
  For dies passing all applied single-port tests:
   Apply March s2PF- via the two ports;
   Apply March d2PF- via the two ports;
```

Fig. 17. Test strategy for 2P memories.

Since each SP test (e.g., March C-) is a single-addressing test as well as March s2PF-, these tests can be merged into a single test. The later has to be applied once via each ports due to the port related faults.

6 Test Results/Fault Probabilities

In order to determine the importance of each FFM, their probabilities of occurrence have been calculated by using two approaches [13]. First, by assuming that all SDs are equal likely (E.L) to occur and then by using *Inductive Fault Analysis (IFA)*. IFA has been used to perform to two different layouts ML and WL, with sizes of 32Kbits and 64 Kbits, respectively. The two layouts implement the same electrical memory circuit of Fig. 1. The results are shown in Table 6. It should be noted that the used IFA tool does not deal with partial opens, i.e., opens with a resistance value $R_{op} < \infty$; therefore, only shorts and bridges (see Section 2) have been considered with IFA. Since DRFs are caused only by opens, their probabilities cannot be determined for ML and WL; it is given in the table as "not applicable (n.a)."

It's clear from the table that the probability of occurrence of a FFM is layout dependent. A FFM which is not realistic for a certain layout can have a considerable probability for another one, e.g., the wCF_{ds}&wIRF has a probability of 0.725% for ML and of 0% for WL (using IFA). That means that, in order to reach a very high fault coverage, a test algorithm designer has to take all FFMs into consideration.

Note that for ML, 94.409% of the faults are 1PFs and 5.591% are 2PFs; while for WL, 2PFs consists of 1.387% (that is $\simeq 4$ times smaller than 2PFs for ML). Versions of the two march tests, March s2PF and March d2PF2, have been implemented at Intel. The tests have been applied to 2P memory dies (with ML layout as well as with WL layout) that pass all Intel SP memory tests. The test results show that, from 33,830 good ML dies (i.e., passing all SP tests), 23 dies fail to pass the implemented tests: seven fail to pass March 2PF1 and 21 fail to pass March d2PF2; note that five dies fail to pass both tests. That means that the tests detect 0.0678% of the dies passing all SP tests; which corresponds with a level of 680 Defects per Million (DPM). Moreover, the test results show that, from 2,165,868 good WL dies, 305 dies fail for 2P tests; which corresponds with a level of 141 DPM. (that is $\simeq 4.8$ times smaller than DPM level for ML). It is interesting to note that the fault coverage of the 2P tests for ML and WL is proportional with the probabilities of occurrence of 2PFs, as predicted by IFA.

TABLE 6
Probabilities of FFMs

FFM	SD E.L	IFA ML	IFA WL
SAF	29.921	29.636	10.737
TF	10.697	16.513	24.035
RDF	5.418	11.561	13.041
DRDF	0.623	0.141	0.048
IRF	7.517	11.264	25.336
RRF	2.606	1.275	1.667
NAF	1.048	1.795	6.460
USF	0.944	9.071	5.785
DRF	0.598	n.a	n.a
Total 1PF1s	59.372	81.256	87.109
CF_{ds}	6.819	5.227	5.661
CF_{ir}	3.758	5.657	3.051
CF_{rr}	0.681	2.245	1.221
CF_{st}	11.805	0.022	1.439
CF_{rd}	0.453	0.002	0.089
CF_{dr}	0.101	0.001	0.015
Total 1PF2s	24.456	13.153	11.504
Total 1PFs	84.377	94.409	98.514
wRDF&wRDF	0.554	2.835	0.969
wDRDF&wDRDF	1.573	0.944	0.323
wRDF&wTF	0.371	0.704	0.000
Total 2PF1s	2.498	4.483	1.292
wCF_{ds}&wRDF	3.302	0.118	0.000
wCF_{ds}&wIRF	6.003	0.752	0.000
wCF_{ds}&wRRF	1.485	0.236	0.000
wCF_{ds}&wCF_{ds}	2.234	0.000	0.000
wCF_{rd}&wRDF	0.132	0.001	0.076
wCF_{dr}&wDRDF	0.617	0.001	0.019
Total 2PF2s	13.674	1.108	0.095
Total 2PFs	16.172	5.591	1.387

7 Conclusions

In this paper, realistic fault models for two-port memories have been presented. They are divided into single-port faults (1PFs) and two-port faults (2PFs). The 1PFs can be detected using the conventional SP memory tests, while the 2PFs require special tests; this is because the sensitization of 2PFs requires the use of the two ports simultaneously. The 2PF2s are further divided into 2PFs involving a single cell (2PF1s) and 2PFs involving two-cells (2PF2s). Depending on to which cells the two simultaneous operations are applied (i.e., to the aggressor cell (a-cell) and/or to the victim cell (v-cell)), the 2PFs are divided into $2\mathrm{PF2}_a$s (whereby, both sensitizing operations have to be applied to the a-cell), $2\mathrm{PF2}_v$s (whereby, both sensitizing operations have to be applied to the v-cell), and $2\mathrm{PF2}_{av}$s (whereby, one sensitizing operation has to be applied to the a-cell and one to the v-cell); whereby, the v-cell is the cell where the fault appears, while the a-cell is the cell to which the sensitizing operation(s) (state) should be applied.

Thereafter, a methodology to design tests for the 2PFs has been introduced, resulting in four march tests to detect such faults: March 2PF1, with a test length of $7n$, to detect 2PF1 faults; March $2\mathrm{PF2}_a$, with a test length of $12n$, to detect $2\mathrm{PF2}_a$ faults; March $2\mathrm{PF2}_v$, with a test length of $14n$, to detect $2\mathrm{PF2}_v$ faults, and March $2\mathrm{PF2}_{av}$, with a test length of $10n$, to detect $2\mathrm{PF2}_{av}$ faults. In addition, each of the four march tests has been optimized without negatively impacting the fault coverage. Furthermore, the four march tests have been classified as *single-addressing* tests and *double-addressing* tests. The single-addressing tests, which consist of March 2PF1, March $2\mathrm{PF2}_a$, and March $2\mathrm{PF2}_v$, are tests that access one address at a time (i.e., both ports use the same address); while the double-addressing test, which consist only of March $2\mathrm{PF}2_{av}$ (also named March d2PF), are tests that access two different addresses at a time (i.e., the two ports use different addresses). The single-addressing tests have been merged into a single march test, called March s2PF, with a test length of $16n$; this test is further optimized to $14n$. Note that, in order to detect all realistic 2PFs, only two linear tests have to be used: March s2PF and March d2PF.

Moreover, Inductive Fault Analysis has been applied to two 2P SRAM layouts: ML and WL. The results show that for ML, 94.409% of the faults are 1PFs and 5.591% are 2PFs; while, for WL, 98.514% of the faults are 1PFs and 1.378% are 2PFs. Versions of March s2PF and March d2PF have been implemented at Intel. The test results show that, from 33,830 good ML dies (i.e., passing all SP tests), 23 dies fail to pass the implemented test, i.e., 0.0678%. That corresponds with a Defects per Million (DPM) level of 680 DPM. In addition, they show that from 2,165,868 good WL dies, 305 dies fail for the two 2P tests that corresponds with a level of 141 DPM. Note that the fault coverage of the tests for 2PFs tracks the predicted fault probabilities for 2PFs very well. For obtaining industrial quality levels of less than 100 DPM, tests for 2PFs are indispensable.

Acknowledgments

This work was supported by the Intel Corporation, USA, via grants and internships. The authors express their gratitude to David Eastwick, Mike Rodgers, Mike Spica, and Greg Tollefson from the Intel Corporation for providing extensive information regarding the tools as well as the SRAM design. We also thank them for their helpful comments and suggestions.

References

[1] M. Inoue et al., "A New Test Evaluation Chip for Lower Cost Memory Tests," *IEEE Design and Test of Computers,* vol. 10, no. 1, pp. 15-19, Mar. 1993.

[2] M.J. Raposa, "Dual Port Static RAM Testing," *Proc. IEEE Int'l Test Conf.,* pp. 362-368, 1988.

[3] B. Nadeau-Dostie, A. Sulburt, and V.K. Agrawal, "Serial Interfacing for Embedded-Memory Testing," *IEEE Design and Test of Computers,* vol. 7, no. 2, pp. 52-63, Apr. 1990.

[4] T. Matsumura, "An Efficient Test Method for Embedded Multi-Port RAM with BIST Circuitry," *Records 1995 IEEE Int'l Workshop Memory Technology, Design, and Testing,* pp. 62-67, 1995.

[5] S.W. Wood et al., "A 5Gb/s 9-Port Application Specific SRAM with Built-in Self-Test," *Records 1995 IEEE Int'l Workshop Memory Technology, Design and Testing,* pp. 68-73, 1995.

[6] Y. Wu and S. Gupta, "Built-In Self Testing for Multi-Port RAMs," *Proc. Sixth Asian Test Symp.,* pp. 398-403, 1997.

[7] V.C. Alves and M. Nicolaidis, "Detecting Complex Coupling Faults in Multi-Port RAMs," IMAG Research Report No. RR978, Feb. 1991.

[8] M. Nicolaidis, V.C. Alves, and H. Bederr, "Testing Complex Coupling Faults in Multi-Port Memories," *IEEE Trans. VLSI Systems,* vol. 3, no. 1, pp. 59-71, Mar. 1995.

[9] A.J. van de Goor and S. Hamdioui, "Fault Models and Tests for Two-Port Memories," *Proc. 16th IEEE VLSI Test Symp.,* pp. 401-410, April 1998.

[10] S. Hamdioui and A.J. van de Goor, "Consequences of Port Restrictions on Testing Two-Port Memories," *Proc. Int'l Test Conf. (ITC '98),* pp. 63-72, Oct. 1998.

[11] S. Hamdioui and A.J. van de Goor, "Port Interference Faults in Two-Port Memories," *Proc. Int'l Test Conf. (ITC '99)* pp. 1001-1010, Sept. 1999.

[12] J. Zhao, S. Irrinki, M. Puri, and F. Lombardi, "Detection of Inter-Port Faults in Multi-Port Static RAMs," *Proc. VLSI Test Symp.,* pp. 297-302, 2000.

[13] S. Hamdioui, "Testing Multi-Port Memories: Theory and Practice," technical report, Delft Univ. of Technology, Faculty of Information Technology and Systems, Section of Computer Eng., Delft, The Netherlands, Oct. 2000.

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

[15] M.S. Abadir and J.K. Reghbati, "Functional Testing of Semiconductor Random Access Memories," *ACM Computer Surveys,* vol. 15, no. 3, pp. 175-198, 1983.

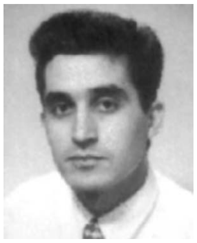
[16] A.J. van de Goor, *Testing Semiconductor Memories, Theory and Practice.* Gouda, The Netherlands: ComTex Publishing, 1998, http://ce.et.tudelft.nl/~vdgoor.

[17] A.J. van de Goor et al., "March LR: Test for Realistic Linked Faults," *Proc. 14th IEEE VLSI Test Symp.,* pp. 272-280, 1996.

[18] R.D. Adams, "Extension of Static Random Access Memories, Modeling and Examination of Pattern for Fault Detection," master of science thesis, Thayer School of Engineering, Dartmouth College, Hanover, New Hampshire, May 1991.

[19] R. Dekker, F. Beenker, and L. Tijssen, "A Realistic Fault Models and Test Algorithms for Static Random Access Memories," *IEEE Trans. Computer-Aided Design,* vol. 9, no. 6, pp. 567-572, June 1990.

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



Said Hamdioui received the baccalaureate degree with honors in 1991 in Morocco. Thereafter, he moved to the Netherlands and joined Delft University of Technology. He received the MS degree (cum laude) in December 1997 and the PhD degree with honors from the same university in October 2001. From October 1998 until July 1999, and from January 2000 until September 2000, he interned with the Intel Corporation at Santa Clara and Folsom, Cailfornia, respectively. He is currently a visiting faculty at the Intel Corporation. His research interests include fault modeling, test design, circuit simulation, DFT, etc. for single-port and multiport SRAMs. He has published more than 15 papers in these areas. He is an associate member of the IEEE.

Ad van de Goor received the MSEE degree from the Delft University of Technology, in Delft, The Netherlands. Additionally, he received the MSEE and PhD degrees from Carnegie-Mellon University, Pennsylvania. He worked for the Digital Equipment Corporation, in Maynard, Massachusetts, as the chief architect of the PDP-11/45 computer, and for IBM in The Netherlands and, in the USA, being responsible for the architecture of the embedded systems. Currently, he is a professor of computer engineering at the Delft University of Technology. His main research interest is testing logic and memories. He has written two books and more than 120 papers in the areas of computer architecture and testing. He is a member of the editorial board of the *Journal of Electronic Testing: Theory and Practice*, and a IEEE fellow.