March SL: A Test For All Static Linked Memory Faults

Said Hamdioui[1,2] Zaid Al-Ars[1] Ad J. van de Goor[1] Mike Rodgers[2]

[1]Delft University of Technology, Faculty of Information Technology and Systems
Computer Engineering Laboratory, Mekelweg 4, 2628 CD Delft, The Netherlands
[2]Intel Corporation, 2200 Mission College Boulevard, Santa Clara, CA 95052
E-mail: S.Hamdioui@ewi.tudelft.nl

Abstract

The analysis of linked faults has proven to be a source for new memory tests, characterized by an increased fault coverage. The paper gives a set of five new tests to target all possible linked faults. The tests are merged into a single test, March SL, detecting all faults in the linked fault space. The preliminary test results of an experiment done at Intel will be reported; they show that March SL scores high and detects some unique faults.

Key words: *Memory testing, linked faults, functional fault models, march tests, fault coverage.*

1 Introduction

Semiconductor memories are an integral part of any VLSI System. Their testing is becoming a major cost factor in the production of the modern VLSIs. Precise fault modeling and efficient test design, in order to keep test cost within acceptable limits, is therefore essential.

One interesting fault modeling tool for memory devices is the concept of *linked faults*, which studies the effect fault models have on each other. Their importance has been validated experimentally by [16], which describes the results of testing 800 DRAM chips with different march tests. March LA [15], designed specifically for detecting linked faults, had a higher fault coverage than the other march tests. However, the design of March LA was based on the fault models which were known at that time. Experimental work [2, 3, 7, 8, 10] has shown the existence of several new fault models.

This paper gives test algorithms for all possible linked faults presented in our previous work [4]; the fault models are validated based on defect injection and circuit simulation. The paper is organized as follows. Section 2 starts with a definition of the linked fault concept. Section 3 briefly discusses the space of linked faults. Section 4 gives the tests. Section 5 presents a theoretical and an industrial evaluation of the newly introduced tests. Section 6 ends with the conclusions.

2 Linked fault concept and definition

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

Simple faults: These are faults which cannot influence the behavior of each other. That means that the behavior of a simple fault cannot change the behavior of another one; therefore *masking* cannot occur. All traditional memory fault models are simple faults; e.g., Transition Faults, Read Destructive Faults, Coupling Faults, etc.

Linked faults: These are faults that do influence the behavior of each other; i.e., the behavior of a certain fault can change the behavior of another one such that *masking* can occur [13, 17]. Note that linked faults consist of two or more simple faults. Due to masking, testing of linked faults is more complicated than testing of simple faults.

Linked faults (LFs) describe an interesting type of faulty behavior that takes place when more than one *fault primitive (FP)* is sensitized. A FP is a mathematical notation describing a single fault [14]; it is represented as $< S/F/R >$, where S *sensitizing operation sequence* (e.g., read 0 from a cell containing 0 (0r0)) that sensitizes a fault F (e.g., cell flips from 0 to 1) in the *victim cell (v-cell)* C_v, and R describes the logic output level (e.g., 1) in case S is a read operation applied to the faulty cell C_v (e.g., a read 0 destructive fault is presented as $< 0r0/1/1 >$).

Assume that the faulty behavior of a memory contains two FPs that share the same v-cell. then FP$_1$ = $<S_1/F_1/R_1>$ **linked** to FP$_2$ = $<S_2/F_2/R_2>$ is denoted as FP$_1$ $\rightarrow$ FP$_2$. This means that the linked fault consists of FP$_1$ *linked to* FP$_2$. If the sensitizing operation sequence (S$_1$) of FP$_1$ is applied first, it sensitizes a fault F_1 in the v-cell, and

1081-7735/03 $17.00 © 2003 **IEEE**

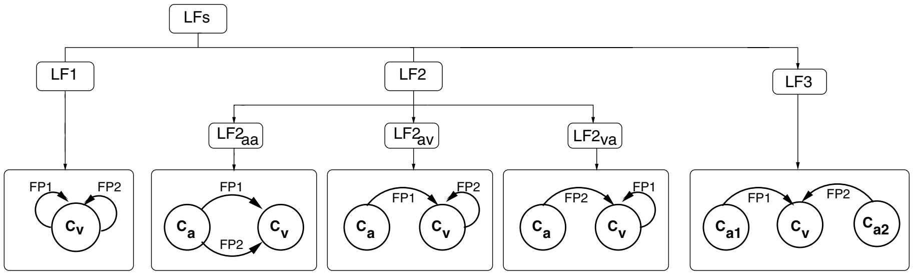

Figure 1. Classification of linked faults

when the S$_2$ of FP$_2$ is applied next, it sensitizes a fault F_2 in the *same v-cell*, but with a fault effect *opposite* to that of the S$_1$ of FP$_1$. The net result is that the fault effect of FP$_2$ *masks* the fault effect of FP$_1$. A FP$_1$ = $<S_1/F_1/R_1>$ is **linked** to FP$_2$ = $<S_2/F_2/R_2>$ if the following three conditions are satisfied [4, 9]:

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 contain.
2. **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.
3. **3. FP$_2$ is compatible with FP$_1$.** This condition applies only in the case S$_2$ of the FP$_2$ is applied immediately after S_1 to 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$_2$.

3 Linked fault space

This section gives a short description of the linked fault space; for more details see [4, 9].

Figure 1 shows the three classes of the LFs based on a combination of two simple FPs. The considered simple FPs are single-cell and two-cell (i.e., coupling) faults. Because the 'linked to' relationship is *not commutative*, the classification also takes the order in which the two FPs are sensitized into consideration.

1. The LFs involving a single cell (LF1s): They are based on a combination of two single-cell FPs. The LF is sensitized by sensitizing the two FPs sequentially in time to the same cell (i.e., v-cell).

2. The LFs involving two cells (LF2s): These are based on a combination of two two-cell FPs, or on a combination of a single-cell FP and a two-cell FP. They are therefore divided into three types:

a. The LF2$_{aa}$: This LF is based on a combination of two two-cell FPs; both FPs have the same *aggressor cell (a-cell)* C_a as well as the same v-cell. The a-cell is the cell to which the sensitizing operation (or state) should be applied in order to sensitize the fault, while the v-cell is the cell where the fault appears; or alternatively, the operation applied to the v-cell will sensitize the fault in the v-cell, provided that the a-cell is in a given state.

b. The LF2$_{av}$: This LF is based on a combination of one two-cell FP$_1$ and one single-cell FP$_2$; whereby the two-cell FP is sensitized first.

c. The LF2$_{va}$: This LF is similar to LF2$_{av}$. However, in this case, first the single-cell FP should be sensitized; thereafter the two-cell FP.

3. The LFs involving three cells (LF3s): They are based on a combination of two two-cell FPs which have different a-cells, but the same v-cell.

The single-cell and the two-cell FPs of Figure 1 can be any possible simple single-cell FPs, respectively any possible simple two-cell FP.

4 Tests for linked faults

This section gives tests for LFs; see Figure 1. The tests for each class have been derived in such a way that first the test conditions have been established, which thereafter have been used to develop test algorithms. A detail description of such tests is given in [9]. Due to lack of space, only the tests of each fault (sub)class will be listed.

4.1 March Tests for LF (sub)classes

The LF space based on two different simple faults is divided into five (sub)classes; see Figure 1. Table 1 lists the march test for each fault (sub)class together with their test lengths (T.L.) as presented in [9].

4.2 Test for all linked faults

Table 2 summarizes the fault coverage of all tests presented in Table 1 [9]. It also shows their required test length

Table 1. March tests for the different linked faults (sub)classes

LF class	March test	Test description	T.L
LF1	March LF1	$\{\Updownarrow(w0);\Updownarrow(r0,r0,w1,w1,r1);\Updownarrow(r1,r1,w0,w0,r0)\}$	11n
LF2$_{aa}$	March LF2$_{aa}$	$\{\Updownarrow(w0);\Uparrow(r0,r0,w1,w1,r1,r1,w0,w0,r0,w1);\Uparrow(r1,r1,w0,w0,r0,r0,w1,w1,r1,w0);$ $\Downarrow(r0,r0,w1,w1,r1,r1,w0,w0,r0,w1);\Downarrow(r1,r1,w0,w0,r0,r0,w1,w1,r1,w0)\}$	41n
LF2$_{av}$	March LF2$_{av}$	$\{\Updownarrow(w0);\Updownarrow(r0,r0,w1,w1,r1);\Updownarrow(r1,r1,w0,w0,r0)\}$	11n
LF2$_{va}$	March LF2$_{va}$	$\{\Updownarrow(w0);\Uparrow(w1,w0,w0,r0,r0);\Downarrow(w1,w1,r1,r1);\Uparrow(w0,w1,w1,r1,r1);\Downarrow(w0,w0,r0,r0)\}$	19n
LF3	March LF3	$\{\Updownarrow(w0);\Uparrow(r0,r0,w1,w1,r1,r1,w0,w0,r0);\Downarrow(r0,r0,w1,w1,r1,r1,w0,w0,r0);$ $\Updownarrow(w1);\Uparrow(r1,r1,w0,w0,r0,r0,w1,w1,r1);\Downarrow(r1,r1,w0,w0,r0,r0,w1,w1,r1)\}$	38n

(T.L.), including the initialization. In the table '+' indicates that the test detects the corresponding LF, and '-' indicates that the test does not cover, or only partially covers, the corresponding LFs. For example, March LF1 detects all LF1 faults and LF2$_{av}$ faults; however, it does not detect (all) LF3 faults. The evaluation of each test is done by verifying whether the test satisfies the detection condition(s) of the corresponding linked fault (sub)class [9].

Table 2. Summary of linked faults tests

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

The table clearly shows that March LF2$_{aa}$ detects all considered linked faults. March LF2$_{aa}$ will be referred from now on as **March SL**: a test for *all static linked faults*, based on two simple faults.

5 Evaluation of March SL

This section first gives a quantitative analysis of the traditional tests regarding their capability of detecting the linked faults discussed in this paper [9]. Thereafter, the preliminary test results of an experiment done on Intel advanced high speed caches will be presented.

5.1 Analytical evaluation

Table 3 summarizes the fault coverage of the most known memory tests for linked faults. The test length of each test is given; n denotes the size of the memory, R denotes the number of rows and C denotes the number of columns. The description of the tests of Table 3 are given below:

1. Scan [1]: $\{\Uparrow(w0);\Uparrow(r0);\Uparrow(w1);\Uparrow(r0)\}$
2. MATS+ [12]: $\{\Updownarrow(w0);\Uparrow(r0,w1);\Downarrow(r1,w0)\}$
3. MATS++ [5]: $\{\Updownarrow(w0);\Uparrow(r0,w1);\Downarrow(r1,w0,r0)\}$
4. March C- [11, 17]: $\{\Updownarrow(w0)$; $\Uparrow(r0,w1)$; $\Uparrow(r1,w0)$; $\Downarrow(r0,w1)$; $\Downarrow(r1,w0)$; $\Updownarrow(r0)\}$
5. PMOVI [6]: $\{\Downarrow(w0)$; $\Uparrow(r0,w1,r1)$; $\Uparrow(r1,w0,r0)$; $\Downarrow(r0,w1,r1)$; $\Downarrow(r1,w0,r0)\}$
6. March SR [8]: $\{\Downarrow(w0)$; $\Uparrow(r0,w1,r1,w0)$; $\Uparrow(r0,r0)$; $\Uparrow(w1)$; $\Downarrow(r1,w0,r0,w1)$; $\Downarrow(r1,r1)\}$
7. Galpat [5]: $\{\Uparrow(w0)$; $\Uparrow_b(w1_b,\Uparrow_{-b}(r0,r1_b),w0_b)$; $\Uparrow(w1)$; $\Uparrow_b(w0_b,\Uparrow_{-b}(r1,r0_b),w1_b)\}$

   Galpat has a time complexity of $O(n^2)$, where n is the size the memory cell array. It starts with initializing the memory cell array to a known data-background (e.g., $\Uparrow(w0)$). During the test, the base cell, which walks through the memory cell array, is written with the data complement value (i.e., $w1_b$). Thereafter all other cells (i.e., *target cells*) are read, but after reading each target cell also the base cell is read (i.e., $\Uparrow_{-b}(r0,r1_b)$). The base cell is finally written to the initial data-background (e.g., $w0_b$) This whole sequence is then repeated with inverted data.
8. Walking 1/0 [17]:
   $\{\Uparrow(w0);\Uparrow_b(w1_b,\Uparrow_b(r0),r1_b,w0_b)$;
   $\Uparrow(w1);\Uparrow_b(w0_b,\Uparrow_b(r1),r1_b,w0_b)\}$

   Walking 1/0 is similar to Galpat; the difference is in reading the base cell. With Walking 1/0, after each step, all cells are read with the base cell last.

In the table, e.g., March C- detects 10 of 12 LF1 possible faults, 10 of 24 LF2$_{aa}$ faults, 11 of 16 LF2$_{av}$ faults, 9 of 18 LF2$_{va}$ faults, and 10 of 24 LF3 faults [9]. The total number of linked faults detected by March C- is then 50 from 94. The table clearly shows that a test with a higher test length does not necessarily have a higher fault coverage for linked faults, and that none of the tests can cover all considered linked faults. Even with all traditional tests of Table 3, a 100% FC of LFs can not be reached.

5.2 Industrial evaluation

The tests shown in Table 3 have been used in an experiment at Intel, and applied to advanced SRAMs chips. The

Table 3. Analytical comparison of the memory tests

#	Tests	Test Length	LF1	LF2$_{aa}$	LF2$_{av}$	LF2$_{va}$	LF3	Total FC
1	Scan	4n	8/12	4/24	8/16	3/18	4/24	27/94
2	Mats+	5n	8/12	0/24	8/16	1/18	0/24	17/94
3	Mats++	6n	10/12	0/24	8/16	4/18	0/24	22/94
4	March C-	10n	10/12	10/24	11/16	9/18	10/24	50/94
5	PMOVI	13n	10/12	15/24	11/16	12/18	15/24	63/94
6	March SR	14n	10/12	16/24	13/16	15/18	18/24	72/94
7	Galpat	6n+4nRC	10/12	13/24	9/16	14/18	13/24	59/94
8	Walking 1/0	8n+2nRC	8/12	12/24	9/16	10/18	12/24	51/94
9	**March SL**	41n	12/12	24/24	16/16	18/18	24/24	94/94

SRAM memory considered in this experiment has a size of 512KBytes, with a word size of $B = 32$ bits, and is tested at 200 MHz.

5.2.1 Used stress combinations

Each test has been applied using different stress combinations. The used stresses are the address orders and the data-backgrounds. The used addressing stresses consist of two type of addressing:

1. Fast X (fx): Fast X addressing is simply incrementing or decrementing the address in such a way that each step goes to the next row.

2. Fast Y (fy): Fast Y addressing is simply incrementing or decrementing the address in such a way that each step goes to the next column.

A *data-background (DB)* is the pattern of ones and zeros as seen in an array of memory cells. The most common types of data-background are four:

1. Solid (s): All 0s, all 1s.
2. Checkerboard (c): 010101.../ 101010
3. Column strip (cs): 000000.../ 111111
4. Row strip (rs): 010101...

5.2.2 Test results

Table 4 lists the 40 total number of tests applied. A *test* consists of a *base test (BT)* (e.g., March C-) applied using a particular *stress combination (SC)*. The total number of tests is 40; that is the number of BTs (i.e., 11) multiplied with the corresponding number of SCs. Note that Galpat and Walking 1/0 are each implemented two times in their $O(n\sqrt{n})$ versions, which restrict the read operation actions only to the same column (e.g., GalColumn), to the same row, or to the same diagonal as the *base cell*. The column '#SC' gives the number of SCs each BTs is used with. A '+' in the table indicates that the corresponding SC is implemented, and a '-' denoted that it is not. E.g., Mats+ is implemented using 'fx-s' and 'fy-s' SCs.

All SCs have been implemented at high power supply voltage because generally testing at high voltage scores better than at low voltage. From the large volume of SRAM chips tested at wafer level and low temperature, 1577 chips failed; 1343 chips do fail all SCs (i.e., 40) while 201 chips do not. From now on, we will only concentrate on SRAM chips that do not fail all SCs since they are the most important.

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

Table 5 shows the *intersections* and the *union* of the 11 BTs. A die belongs to the intersection of two BTs if both BTs found the die to be faulty, and belongs to the union of two BTs if at least one of the two BTs found the die to be faulty. The first column in each table gives the BT number; the second the column name of the BT. The column 'FC' lists the fault coverage of the corresponding BT; the column 'UFs' gives number of *unique faults (UFs)* each BT detects. Unique faults are faults that are only detected once with a single test; e.g., GalRow detects 9 unique faults that are not detected with any other test. The intersection and the union of each pair of BTs is shown in the rest of the table. The numbers on the diagonal give the fault coverage (FC) of the BTs, which are also listed in the column 'FC'. The part above the main diagonal shows the intersection for each BT pair, while the part under the diagonal lists the union of each BT pair; for example, the intersection of March C- and PMOVI is 179 and their union is 185. Based on the data of the table, one can conclude the following:

- Total number of faulty chips detected: 201.
- The best BT, in terms of FC, is March SL. However, this is a strong conclusion since not all BTs used the same SCs.
- There are 15 unique faults, detected by 5 tests listed in Table 6 with their FC.

Table 4. List of used stress combinations

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

Table 5. Intersection and unions of BTs

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

Table 6. BTs detecting unique faults

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

It has been shown that linked faults can not be covered with any of the traditional tests, therefore March SL has been designed to target linked faults. The detected UFs by March SL are (probably) linked faults. Other BTs detecting UFs need more detailed analysis. These BTs detect faults that can not be explained using the well-known fault models. This means that additional fault models and/or fault classes exist. By analyzing each failed test for UFs, new fault models will be introduced; Failure analysis can also be used for better understanding of the underlying defect causing such SF behavior. A new test(s) with a shorter test length, targeting the detected UFs and replacing the non-linear empirical tests, can be constructed and used for further test purposes, instead of expensive ones. Note that the best test in terms of detecting UFs is GalRow. However it detects considerably fewer faults.

- Some BTs detect supersets of faults of other BTs. For example: GalColumn detects a superset of WalkColumn. Table 7 summarizes the BTs detecting supersets of faults of other ones. The initial 11 BTs of Table 4 can therefore be reduced to 7 BTs while achieving the same fault coverage; that means that the BTs covered by other ones can be removed.

Table 7. List of BTs detecting supersets of other ones

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

- The best union pair of BTs, in terms of FC, is the union of March SL and GalRow with FC=195.

5.2.3 Effect of stresses on March SL

Table 8 shows the effect of SCs on March SL. The first column lists the uses SC; e.g., (fx,c) denotes 'fast x' addressing and 'checkerboard data-background'. The rest of the table prints the effect of the impact of the stresses on the FC. The main diagonal, which is printed in bold, gives the FC of each SC; the part above the main diagonal lists the *intersection* of each pair of SCs; while the part below the diagonal lists the *union*of each pair of SCs. Remember that the maximal FC that have been achieved with all SCs for March SL is FC=185.

Table 8. SCs intersections and unions for March SL

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

Based on the data of Table 8, the following can be concluded:

- The (fy,cs) is the most effective SC detecting 182/185.
- The union of (fy,cs) and (fy,rs) is the most effective SC pair union detecting 184/185 faults.
- The (fx,cs) covers the superset of (fx,rs); and the (fy,cs) covers that of (fy,c).
- An analysis (not shown here) reveals that a least three SCs are required in order to detect 185/185 faults; these can be one of the following three triple of SCs: [(fx,cs),(fx,s),(fy,rs)], [(fx,cs),(fx,s),(fy,s)], or [(fx,s),(fy,cs),(fy,rs)],

6 Conclusions

In this paper a brief description of linked fault has been presented. This space has been divided, based on the number of cells required in the LF, into five different fault (sub)classes. Thereafter,a set of five linear march tests, each for each fault (sub)class, has been presented. An analytical evaluation of these tests reveals that one of the tests has also the capability to detect all other linked fault (sub)classes. This test, which is named as *March SL*, can be thus used to cover all single-port, static, linked faults in RAMs.

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

References

[1] M.S. Abadir and J.K. Reghbati, "Functional Testing of Semiconductor Random Access Memories", *ACM 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 Pro. of Asian Test Symposium*, pp. 282-289 , 2000.

[4] Z. Al-Ars, S. Hamdioui and Ad J. van de Goor, "A Fault Primitive Based Analysis for RAM Linked Faults", *To Appear in Proc. of IEEE International Workshop on Memory Technology, Design and Test,* 2003.

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

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

[7] R. Dekker, F. Beenaker, and L. Thijssen, " A realistic Fault Model and Test Algorithm for Static Random Access Memories," *IEEE Trans. on Computers*, V. 9, pp. 567-572, 1990.

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

[9] S. Hamdioui, Z. Al-ars, A.J. van de Goor and M. Rodgers, "Linked Faults in Random Access Memories: Concept, Fault Models, Test Algorithms and Industrial Results", *To Appear in IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, 2003.

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

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

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

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

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

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

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

[17] A.J. van de Goor, *Testing Semiconductor Memories, Theory and Practice*, ComTex Publishing, Gouda, The Netherlands, 1998.