PPM Reduction on Embedded Memories in System on Chip

Said Hamdioui Zaid Al-Ars
Delft University of Technology
Computer Engineering Laboratory
Mekelweg 4, 2628 CD Delft, The Netherlands
{S.Hamdioui, Z.Alars}@ewi.tudelft.nl

Javier Jimenez Jose Calero
Design of Systems on Silicon (DS2)
Calle Charles Robert Darwin, 2
46980, Paterna, Valencia, Spain
{Javier.Jimenez,Jose.Calero}@ds2.es

Abstract

This paper summarizes advanced test patterns designed to target dynamic and time-related faults caused by new defect mechanisms in deep-submicron memory technologies. Such tests are industrially evaluated together with the traditional tests at "Design of Systems on Silicon (DS2)" in Spain in order to (a) validate the used fault models and (b) investigate the added value of the new tests and their impact on the PPM level. The preliminary silicon results are presented and analyzed. They validate some of the new dynamic fault models and show the importance of considering dynamic faults for high outgoing product quality.

Key words: *memory testing, static faults, dynamic faults, PPM reduction.*

1 Introduction

The scaling of technology feature sizes gives rise to new defect mechanisms that behave differently at the functional level than traditional ones [1]-[4]. For memories, this means that the traditional test patterns designed to target the traditional (i.e., static) fault models are not anymore sufficient to achieve an acceptable PPM (Part-Per-Million) level. Many new dynamic and time-related faults are developed to model the defect mechanisms in deep submicron memory technologies (e.g., partial opens, cross talk, etc.), and advanced test solutions to cover such faults are also proposed [5]-[14]. However, there is still lack of published industrial data to seriously evaluate the added value of the new fault models and tests and their impact on the PPM level.

This paper gives a classification of recently introduced dynamic faults. In addition, it presents a test(s) for each class. Such tests are thereafter industrially evaluated by applying them to 0.13μm embedded memories of DS2 products. The silicon results are then presented and analyzed.

The paper is organized as follows. Section 2 gives a short overview of traditional/static faults and advanced/dynamic faults and classifies them. Section 3 describes the stress combinations usually used when applying memory tests; such combinations are very critical when detecting dynamic faults. Section 4 gives advanced test solutions to cover the dynamic faults. Section 5 presents an overview of the whole to-be-used tests and stresses in the memory test experiment. Sections 6 and 7 discuss and analyze the silicon results. Section 8 concludes the paper.

2 Static versus dynamic faults for RAMs

Traditional memory fault models referred to as static faults are generally speaking divided into three types [15]:

1. Static Memory Cell Array Faults (sMCAFs): these consist of faults occurring in the memory array; they can be either single-cell faults (e.g., State Fault, Transition Fault, Read Destructive Fault, etc.) or coupling faults (CFs) (e.g., State CF, Transition CF, Disturb CF, etc.) [15]-[25].
2. Static Address Decoder Faults (sADFs): these are faults occurring in the address decoders. They consist of four known fault models [15], [26]-[31]: No-Access sADF, Multiple-cell sADF, multiple-address sADF and other-cells sADF.
3. Static Peripheral Circuit Faults (sPCFs): these are faults occurring in the rest of the memory circuit (e.g., sense amplifier, pre-charge circuits, etc.). These faults are modeled as Stuck-at-faults and bridging faults and considered to be covered with any test detecting sMCAFs. They are thus mapped into sMCAFs [15].

To detect the above three types of static faults, many test algorithms have been introduced with different degrees of success [15]-[30].

With the scaling of technology, new defect mechanisms take place in addition to the known traditional ones. The new defect mechanisms cause fault behaviors that are different from the static faults. They cause mainly time-related faults referred to as dynamic faults. Dynamic faults can also be divided into three types:

12th IEEE European Test Symposium (ETS'07)
0-7695-2827-9/07 $20.00 © 2007 IEEE

1. Dynamic Memory Cell Array Faults (dMCAFs): these are dynamic faults that occur in the memory array. They require more than one operation to be applied to the victim-cell and/or to the aggressor cell in order to sensitize the fault in the victim cell. Example of such faults are: Dynamic Read Destructive Fault, Dynamic Transition Fault, Dynamic Write Disturb Fault, Dynamic Coupling fault, etc. [5]-[10].
2. Dynamic Address Decoder Faults (dADFs): these consist of delay faults that can occur in the address decoders. They are mainly caused by partial opens. Such faults are modeled as [9, 10, 11, 12, 13]:
   (a) Activation delay fault (ActD)
   (b) De-activation delay fault (DeactD)
3. Dynamic Peripheral Circuit Faults (dPCFs): these are time-related faults caused by defects occurring in the peripheral circuits of the memory system like partial opens, partially filled vias and leakages. Dynamic peripheral circuit faults consists of [14, 16]:
   (a) Slow Write Driver Faults (SWDF)
   (b) Slow Sense Amplifier Fault (SSAF)
   (c) Slow PRecharge Circuit Fault (SPRF)
   (d) Bit Line Imbalance Fault (BLIF)

3 Overview of stress combinations

When testing, each test is applied using several stresses. The stresses can be divided into algorithm stresses and non-algorithm stresses. A *non-algorithm stress*, also referred to as an *environmental* stress, specifies the environmental values, such as the supply voltage, the temperature, the timing (the clock frequency), etc.; they are effective during the application of the test.

An *algorithm stress* specifies the way the test is performed, and therefore it influences the sequence and/or the type of the memory operations. The most known algorithm stresses are the address directions (ADs) and the data-backgrounds (DBs).

The mainly used ADs consist of 'fx' and 'fy'.

- 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.
- 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 DB is defined as the pattern of ones and zeros as seen in an array of memory cells. The used DBs are:

- Solid (sDB): all 0s and all 1s.
- Checkerboard (bDB): 0101.../1010.../0101.../1010..
- Column stripe (cDB): 0101.../0101.../0101.../0101.
- Row stripe (rDB): 0000.../1111.../0000.../1111....

4 Advanced tests for dynamic faults

Recently many test algorithms have been developed to target the dynamic fault class of memory faults [7, 8, 10, 13, 14]; see the first block of Table 1.

1. Tests for dynamic memory cell array faults: the framework of all possible dynamic memory cell array faults is presented in [6]. However, only a subset of such space has been shown to be realistic based on defect injection and circuit simulation; mainly some single-cell dynamic faults. March DFr shown in Table 1 [7, 8] is an effective test to target the realistic dynamic memory cell array faults shown to exist in real designs so far.
2. Tests for dynamic address decoder faults: March dADF show in Table 1 is an optimal and efficient test capable of detecting all dADFs [12, 13]. The test has to use the hamming addressing with hamming distance between two addresses of H=1. H is defined as the number of bit positions in which two successive addresses (say Ax and Ay) of an address pair differs. In addition, the test has to be applied using the sDB together with address direction fx (to detect dADFs in row decoder) and fy (to detect dADFs in column decoder).
3. Tests for dynamic peripheral circuits faults. The minimal test set to target all dPCFs is given in Table 1; it consists of two tests [14]: March dPCF$_w$ (w=walking) to detect BLIF and SWDF, and March dPCF$_m$ (m=marching) to detect SSAF and SPRF. Both tests have to be used with fx addressing and sDB or cDB.

5 List of used tests and stresses

Table 1 lists all used tests in the experiment together with their test length and the SCs they will be used with; in the table n denotes the memory size and B the word size. The set of tests is classified into three classes:

1. Dynamic Tests: These consists of all tests shown in first block of Table 1 and explained in Section 4.
2. Static/Classical tests: consist of well know tests that have been used in the industry. Such tests consist of e.g., Scan [32], MATS+ [33], MATS++[34], March C- [35], PMOVI [36], March B and March G [18]. All such tests are used with both ADs (fx and fy) and four DBs (sDB, bDB or cDB and rDB).
3. Special Tests: These consist of three extra tests that are added, based on our experience in the field of memory testing, to detect some specific faults or some unmodeled faults; these tests are: March SAM, March Gal9R and a new March SZ.

Table 1. Set of tests used for the industrial evaluation

Dynamic Tests				
Name	Test length	Description	DB	AD
March DFr	$22n$	$\{\Uparrow(w0);\Uparrow(r0,w0,r0,w1,r1);\Uparrow(r1,w1,r1,w0,r0);$ $\Downarrow(r0,w0,r0,w1,r1);\Downarrow(r1,w1,r1,w0,r0);\Uparrow(r0)\}$	cDB and rDB	fx and fx
March dADF	$\frac{9}{2}n+$ $9\cdot n\cdot\log_2(n)$	$\{\Updownarrow(w0);\Uparrow^{H1}(r0,w1);\Uparrow^{H1}(r1,w0);\Downarrow^{H1}(r0,w1);\Downarrow^{H1}(r1,w0)\}$	sDB	fx and fy (H1 addressing)
March dPCFw	$8n$	$\{\Updownarrow(w0);_X\Updownarrow(w1,r1,w0);\Updownarrow(w1);_X\Updownarrow(w0,r0,w1)\}$	sBd or cDB	fx
March dPCFm	$5n$	$\{\Updownarrow(w0);_X\Updownarrow(r0,w1);_X\Updownarrow(r1,w0)\}$	sDB or cDB	fx
Static/Traditional Tests				
Scan	$4n$	$\{\Uparrow(w0);\Uparrow(r0);\Uparrow(w1);\Uparrow(r1)\}$	all DBs	fx and fy
MATS+	$5n$	$\{\Uparrow(w0);\Uparrow(r0,w1);\Downarrow(r1,w0)\}$	all DBs	fx and fy
MATS++	$6n$	$\{\Updownarrow(w0);\Uparrow(r0,w1);\Downarrow(r1,w0,r0)\}$	all DBs	fx and fy
March C-	$10n$	$\{\Updownarrow(w0);\Uparrow(r0,w1);\Uparrow(r1,w0);\Downarrow(r0,w1);\Downarrow(r1,w0);\Updownarrow(r0)\}$	all DBs	fx and fy
PMOVI	$13n$	$\{\Downarrow(w0);\Uparrow(r0,w1,r1);\Uparrow(r1,w0,r0);\Downarrow(r0,w1,r1);\Downarrow(r1,w0,r0)\}$	all DBs	fx and fy
March B	$17n$	$\{\Updownarrow(w0);\Uparrow(r0,w1,r1,w0,r0,w1);\Uparrow(r1,w0,w1);$ $\Downarrow(r1,w0,w1,w0);\Downarrow(r0,w1,w0)\}$	all DBs	fx and fy
March G	$23n$	$\{\Updownarrow(w0);\Uparrow(r0,w1,r1,w0,r0,w1);\Uparrow(r1,w0,w1);\Downarrow(r1,w0,w1,w0);$ $\Downarrow(r0,w1,w0);\Updownarrow(r0,w1,r1);\Updownarrow(r1,w0,r0)\}$	all DBs	fx and fy
Specific Tests				
March SAM	$[1+28\log_2(B)]\frac{n}{B}$	The description depends on the size of the B; see [37]	*Special DBs*	fx
Gal9R	$38n$	$\{\Uparrow(w0);\Uparrow_b(w1_b,\square(r0,r1_b),w0_b);\Downarrow(w1);\Downarrow_b(w0_b,\square(r1,r0_b),w1_b)\}$	sDB	fy
March SZ	$[1+\log_2(B)]\cdot 7\cdot\frac{n}{B}$	$\{\Downarrow(w0);\Uparrow(r0,w1,r1);\Downarrow(r1,w0,r0)\}$	*Special DBs*	fx and fy

March SAM [37] is developed to target intra-word coupling faults, which are coupling faults that take place between memory cells that belong to the same memory word, and faults due to the interference between I/O paths of the same word. March SAM requires the use of fx AD and *special DBs* that are different from the standard ones; see for more details [37]. The test length is $[1+28*\log_2(B)]*\frac{n}{B}$ where B denotes the word size.

March Gal9R is a simplified from of Galpat [15] where the read actions are restricted only to the eight physical neighbors of the base cell. In the table, '$_\square$' denotes the eight neighbor cells of any base cell and '$_b$' denotes the base cell. For example, '$_\square(r0,r1_b)$' means apply read 0 to the 8 neighbor cells, and after each read apply read 1 to the base cell. The test has to be used with sDB and fy AD.

March SZ is expected to detect interferences between I/O paths, timing related faults (not yet fully understood), cell stability faults and leakage currents. Such a test is repeated with $[1+\log_2(B)]$ DBs which are generated as follows. For B=2, there are 2 DBs:00 and 01; for B=4, there are 3 DBs: 0000, 0101 and 1100; etc. The test length of March SZ is $[1+\log_2(B)]\cdot 7\cdot\frac{n}{B}$.

6 Silicon results

The 14 test algorithms with their associated algorithmic stresses (see Table 1) were implemented at the same non-algorithmic stresses (typical voltage, typical speed and room temperature). The total number of resulting tests is then 54 (i.e., 42 static tests, 8 dynamic tests and 4 specific tests). The tests were applied two times to a huge number of $0.13\mu m$ embedded SRAMs of 256KB at DS2, and the 'device under test' is considered faulty by a test if it fails the same test twice.

The total number of failing devices is 20511, from which 14373 failed all 54 tests (i.e., static, dynamic and specific). The latter failures consist of the easy to detect static faults like static single cell faults. From now on, we will focus only on the chips that do not fail all the tests, as they are the most interesting. These consist of 6138 (= the total failed chips - the chips failing all tests = 20511-14373).

Figure 1 shows the venn-diagram of the 6138 failing devices for the different test classes. The Static Tests detect 5473, the Dynamic Tests detect 1985, whereas the Specific Tests detect 2327. Note that 1579 faults are detected by all the three test classes, but not by all 54 tests. For example, a fault is detected by Dynamic Tests since March dADF detects the fault, but not by March dPCFw neither by March dPCFw. From the venn-diagram we can conclude the following:

- The fault coverage (i.e., the number of detected defective chips) achieved with Static Tests is the highest. This can be explained by the high number of static tests (i.e., 42 tests) as compared with the total number of tests of the other two test classes (i.e., 12 tests).
- The number of faults that are *only* detected with Static Tests is 3362. These faults are not modelled as static faults since March DFr (designed to target some dy-

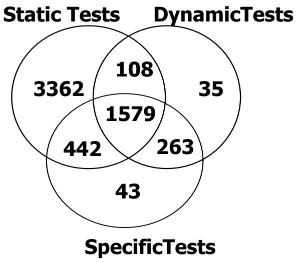

Figure 1. Venn-diagram of failing chips

namic faults), which also cover all static faults [7, 8], does not detect them. This clearly indicates that the existing static fault models cannot be used to explain the detected faults, and that additional faults exit (that still need to be understood and modeled).

- The Dynamic Tests detect 35 faults that are neither detected with Static Tests nor with Specific Tests. This may indicate the importance of considering dynamic faults in order to achieve a high product quality. Dynamic faults are becoming very important due to the following: (a) the continued increase of memory size leads to long signal lines (e.g., word line and bit lines), which means more coupling and cross talking; (b) the increase of the clock speeds makes design more sensitive to time-related faults; (c) the use of copper wiring shifts the predominant failure mode from shorts and bridges to (partial) opens, which behaves as timing related faults, and (d) the constant field scaling has as consequences lower noise margin and high leakage current (due to supply and threshold voltage reduction). In short, faults caused by the new defect mechanism resulting in delay/time related and dynamic faults are becoming a dominant failure mode.

- Specific Tests detect 43 faults that are neither detected with Static Tests nor with Dynamic Tests. Specific tests target mainly intra-word coupling faults, faults due to the interference between I/O paths of the same word, and cell stability faults. Such faults are partially understood and modeled. A detailed analysis of such faults will be useful for further understanding of the defect mechanisms and the appropriate way to model them at the functional level.

7 Discussion and analysis

This section discusses and analyzes the results of Dynamic Tests and Specific Tests in order to understand their added values and validate the dynamic fault models.

Table 2. Union/ intersection of Dynamic Tests

#	Test	UF	1	2	3	4
1	March DFr	15	**1879**	1945	1936	1933
2	March dADF	1	1148	**1214**	1304	1277
3	March dPCFw	4	352	319	**409**	1079
4	March dPCFm	0	888	879	272	**942**

Table 3. UFs detected by Dynamic Tests

Test(s)	# UFs
March DFr	15
March dADF	1
March dPCFw	4
March dPCFm	0
March DFr and March dADF	2
March DFr and March dPCFw	0
March DFr and March dPCFm	1
March dADF and March dPCFw	0
March dADF and March dPCFm	0
March dPCFm and March dPCFw	1
March DFr, March dADF and March dPCFw	0
March DFr, March dADF and March dPCFm	9
March dADF, March dPCFw and March dPCFm	2
Total	35

7.1 Dynamic Tests

Table 2 shows the unions and the intersections of the four dynamic test algorithms. A defective die belongs to the union of two tests if at least one of the tests found the die to be faulty, and belongs to the intersection of two tests if both tests found the die to be faulty. The first column in the table gives the test number; the second column the name of test algorithm, the column UF lists the *unique failures (UFs)* each test detects. Unique failures are faults that are only detected once with a single test; e.g., March DFr detects 15 UFs that are not detected by any other test used in the experiment. The numbers on the diagonal (printed in bold font) give the fault coverage (FC) of the tests; e.g., March dADF has a FC=1214. The FC is defined here as the number of chips detected to be faulty. The part above the main diagonal shows the union for each test pair, while the part under the diagonal lists the intersection of each test pair; for example, the union of March DFr and March dPCFw is 1936 and their intersection is 352. Referring back to Figure 1, we can see that there are 35 faults that are only detected by Dynamic Tests. Table 3 shows the distribution of such faults on the dynamic tests. For example, March dPCFm and March dPCFw detect one such fault; that means that such single fault is only detected by both March dPCFm and March dPCFw.

Some important conclusions based on the above are the following:

- March DFr scores the highest in terms of FC and the number of UF as compared with the other dynamic tests. That can be explained by the fact the occurrence probability of faults in the memory array is very high as compared with the rest. Typically more than 70% of the faults are related to the memory array. Note that March DFr is designed mainly to target dynamic faults in the memory array, while the other tests (March dADF, March dPCFw and March dPCFm) are designed to target dynamic faults in the address decoders and the peripheral circuits.

- March dADF detects only one UF. A detailed analysis of such failure, which has a partial column bitmap signature, has been done in order to diagnose the root cause behind it and to validate the dynamic fault model for address decoders (i.e., Activation and De-activation delay [10, 11, 12, 13]). The results revealed that the fault was caused by a *partially filled via* in the very last stage of the column decoder impacting the timing of the ”Column Select” signal of the memory.
- March dPCFw and March dPCFm have the lowest FC as it could be expected; they designed to target faults in the peripheral circuits only. However, March dPCFw detects 4 UFs. A detailed analysis of couple of samples of such failures have been done while focusing on the target faults by March dPCFw. The preliminary results reveal that for a fault with a partial column signature, there is some irregularities in the pass transistors of the victim cells. This may increase the leakage current during the read operation. With decreasing feature widths, the transistors increasingly draw more current in the off-state. This also applies to the pass transistors that impact the possibility of reading the correct value of a cell. Leakage current during read operations has three components [38]: (a) sub-threshold leakage, (b) gate leakage current and (c) junction leakage current.
- Unique faults distribution of Table 3 shows that about 9 UFs are detected by all Dynamic Tests except March dPCFw. This means that these UFs can be either located in memory cell array, the address decoders or in the peripheral circuits (sense amplifier or pre-charge circuit); this makes it not possible to distinguish between the detected faults and therefore difficult to diagnose the location and the cause of such defects. In this case, using the so called ”*Test Primitive*” concept (which only targets a single *Fault Primitive* at a time) can be very helpful.

7.2 Specific Tests

Table 4 shows the unions and the intersections of the three specific tests; see also Table 1. The representation used in Table 4 is similar to that used in Table 2. Referring back to Figure 1, we can see that there are 43 faults that are only detected by Specific Tests. Table 5 shows the distribution of such faults on the three tests. Some important conclusions:

Table 4. Union/ intersection of Specific Tests

#	Test	UF	1	2	3
1	March SAM	6	**1484**	2184	2145
2	GAL9R	4	989	**1689**	2291
3	March SZ	5	1423	1482	**2084**

Table 5. UFs detected by Specific Tests

Test(s)	# UFs
March SAM	6
GAL9R	4
March SZ	5
March SAM and GAL9R	1
March SAM and March SZ	22
GAL9R and March SZ	2
March SAM and GAL9R and March SZ	3
Total	43

- March SAM scores the highest in terms the number of UF as compared with other Specific Tests. March SAM is designed to target intra-word coupling faults and faults due to the interference between I/O paths of a single word. But, the test also detect the majority of static faults [37]. This can explain its high FC. In addition, the UFs are suspected to be faults due to the coupling between (wires associated to) cells belonging to the same word and/or interference/cross-talk between the I/O paths.
- GAL9R is a short version of Galpat [15]. The latter is an ad hoc test because when designed it was not based on any fault models; it was rather based on speculations. However, the test is used a lot in the industry and in different versions. This indicates its added value for achieving a high product quality. The type of detected faults and the defect mechanisms behind them are just partially understood. Galpat is usually used with the following intentions: (a) to deal with cell stability problems, (b) to cover the speed-related faults (e.g., in the address decoders), (c) to detect defects that degrade the cell stability.
- March SZ scores the highest in terms of FC as compared with the other Specific Tests. March SZ is designed to target faults due to interferences; that it is also what March SAM is supposed to detect (partially) as well. This explains the high intersections (i.e., 1423) of the two tests, while the FC of the two tests are 1484, respectively, 2084. In short the March SAM and March SZ target sets of faults that are predominantly the same. The fact that the number of detected UFs (see Figure 5) by the two tests is 22 validates this statement as well.

8 Conclusions

In this paper an overview of existing fault models for embedded memories has been given, classified into static and dynamic faults. Dynamic faults are more advanced fault models intended to cover the new defect mechanisms in deep-submicron memory technology. A set of advanced test patterns to target the dynamic faults has been given. Such tests have been industrially evaluated in order to validate the dynamic fault models and evaluate the added value of the tests and their impact on the PPM level. Based on the preliminary results, we can draw the following conclusions:

- Some dynamic fault models have been validated based on failure analysis. E.g., March dADF designed to target dynamic faults in the address decoders detected a unique fault that appears to be due a partially filled via in the last last stage of the column decoder impacting the timing of the ”‘Column Select” signal.
- Using dynamic tests increases the quality of the outgoing products and reduces the number of escapes. In the limited performed experiment, 35 unique faults (from a total of 20511) were detected by dynamic tests. This is can be translated into a large number of PPM if not taken into consideration.
- There are many things that are still not understood. A large number of faults detected only by static tests can be explained with the existing fault models. Such faults may belong to another class and are detected by static tests since they are used with many stress combinations (addressings and data-backgrounds). This indicates the importance of performing a more detailed analysis to understand the defect mechanisms and model them in an appropriate way.

References

[1] B. Vermeulen, C. Hora, B. Kruseman, E.J. Marinissen and R. van Rijsinge, ”Trends in Testing Integrated Circuits,” in Proc. IEEE Int’l Test Conf., 2004, pp. 688-697.

[2] F. Caignet, S. Delmas-Bendhia and E. Sicard, ”The Challenge of Signal Integrity in Deep-Submicrometer CMOS Technology,” in Proc. IEEE, 2001, vol. 89, no. 4, pp. 556-573.

[3] S. Kundu, T.M. Mak and R. Galivanche, ”Trends in Manufacturing Test Methods and Their Implications,” in Proc. IEEE Int’l Test Conf., 2004, pp. 679-688.

[4] S. Hamdioui, G. Gaydadjiev and A.J. van de Goor, ”The State-of-art and Future Trends in Testing Embedded Memories,” in Proc. IEEE Int’l Workshop on Memory Technology, Design and Testing, 2004, pp. 54-59.

[5] L. Dilillo et al., ”Comparison of Open and Resistive-Open Defect Test Conditions in SRAM Address Decoders,” Proc. IEEE Asian Test Symp., pp. 250-255, 2003.

[6] Z. Al-Ars, A. J. van de Goor, Static and Dynamic Behavior of Memory Cell Array Spot Defects in Embedded DRAMs, IEEE Transactions on Computers, pp. 293-309, March 2003

[7] S. Hamdioui, Z. Al-Ars, and A.J. van de Goor, ”Testing Static and Dynamic Faults in Random Access Memories,” Proc. Of IEEE VLSI Test Symposium, 2002, pp. 395-400.

[8] G. Harutunyan, V.A. Vardanian and Y. Zorian, ”Minimal March Tests for Dynamic Faults in Random Access Memories”, In Proc of ETS, pp. 43 - 48, 2006

[9] M. Azimane and A.K. Majhi, ”New Test Methodology for Resistive Open Defect Detection in Memory Address Decoders”, Proc. IEEE VLSI Test Symp., pp. 123-128, 2004

[10] L. Dilillo, P. Girard, S. Pravossoudovitch, A. Virazel, S. Borri, M Hage-Hassan, ”Dynamic read destructive fault in embedded-SRAMs: analysis and march test solution”, In proc. Of IEEE European Test Symposium, pp. 140 - 145, 2004.

[11] J. Otterstedt et al., ”Detection of CMOS Address Decoder Open Faults with March and Pseudo Random Memory Tests”, Proc. IEEE Int’l Test Conf., pp. 53-62, 1998

[12] M. Sachdev, ”Open Defects in CMOS RAM Address Decoders”, IEEE Design and Test of Computers, pp. 26-33, Apr.-June 1997.

[13] S. Hamdioui, Z. Al-ars, A.J. van de Goor, ”Opens and Delay Faults in CMOS RAM Address Decoders”, IEEE Trans. Computers, Vol. 55, No. 12, pp. 1630-1639, Nov., 2006.

[14] A.J. van de Goor, S. Hamdioui, and R. Wadsworth, ”Detecting Faults in the Peripheral Circuits and Evaluation on SRAM Tests”, Proc. IEEE Int’l Test Conf., pp. 114-123, 2004.

[15] A.J. van de Goor, ”Testing Semiconductor Memories, Theory and Practice”, ComTex Publishing, Gouda, The Netherlands, (second edition), 1998.

[16] R.D. Adams and E.S. Cooley, ”False Write Through and Un-Restored Write Electrical Level Faults Models for SRAMs”, In Proc. Of lEEE Inr. Workshop on Memory Technology, Design and Test, pp. 27-32, 1997

[17] R.D. Adams and E.S. Cooley, ”Analysis of Deceptive Destructive Read Memory Fault Model and Recommended Testing”, Records North Atlantic Test Workshop, pp. 27-32, May 1996.

[18] D.S. Suk and S.M. Reddy, ”A March Test for Functional Faults in Semiconductors Random-Access Memories”, IEEE Transactions on Computers, C-29(6), pp. 419-429, 1980.

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

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

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

[22] R. Dekker F. Beenker, and H. Thijssen, ”A Realistic Fault Models and Test Algorithms for Static random Access Memories”, IEEE Trans. on Computers,C9(6), pp. 567-572, 1990.

[23] M.S. Abadir and J.K. Reghbati, ”Functional Testing of Semiconductor Random Access Memories”, ACM Computer Surveys, 15(3), pp. 175-198, 1983.

[24] S. Hamdioui, A.J. van de Goor, and M. Rodgers, ”March SS: A Test for All Static Simple RAM Faults”, Proc. IEEE Int’l Workshop Memory Technology, Design and Testing, pp. 95-100, 2002.

[25] S. Hamdioui, and A.J. van de Goor, ”Experimental analysis of spot defects in SRAMs: realistic fault models and tests”, Proc. Ninth Asian Test Symp., 2000, pp. 131-138

[26] M.T. Fragano, J.H. Oppold, M.R. Ouellette, and J.P. Rowland, ”Self-Test Pattern to Detect Stuck-Open Faults”, US Patent No. 6,442,085 B1, 27 Aug. 2002.

[27] E. Gizdarsky, ”Detection of Delay Faults in Memory Address Decoders, ”J. Electronic Testing: Theory and Applications”, vol. 16, pp. 381-387, 2000.

[28] B. Nadeau-Dostie et al., ”Serial Interfacing for Embedded-Memory Testing”, IEEE Design and Test of Computers, vol. 7, no. 2, pp. 52-63, 1990.

[29] S. Nakahara et al., ”Built-In Self-Test for GHz Embedded SRAMs Using Flexible Pattern Generator and New Repair Algorithm”, Proc. IEEE Int’l Test Conf., pp. 301-310, 1999.

[30] J. Otterstedt et al., ”Detection of CMOS Address Decoder Open Faults with March and Pseudo Random Memory Tests”, Proc. IEEE Int’l Test Conf., pp. 53-62, 1998.

[31] M. Sachdev, ”Open Defects in CMOS RAM Address Decoders”, IEEE Design and Test of Computers, pp. 26-33, Apr.-June 1997.

[32] M.S. Abadir and J.K. Reghbati, ”Functional Testing of Semiconductor Random Access Memorie”, ACM Computer Surveys, 15(3), pp. 175-198, 1983.

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

[34] M.A. Breuer and A.D. Friedman, Diagnosis and Reliable Design of Digital Systems, Computer Science Press, Woodland Hills, CA, USA, 1976.

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

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

[37] S. Hamdioui and J.D. Reyes, ”New Data-Background Sequences and Their Industrial Evaluation for Word-Oriented Random-Access Memories”, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 24, No. 6, pp. 892-904, June 2005.

[38] S. Mukhopadhyay, H. Mahmoodi and K. Roy, ”Modeling of Failure Probability and Statistical Design of SRAM Array for Yield Enhancement in Nanoscaled CMOS”’, IEEE Trans. on CAD, vol. 24, No. 12, pp. 1859-1880, December 2005.