Investigation of Single-Cell Dynamic Faults in Deep-Submiron Memory Technologies

Said Hamdioui[1] Zaid Al-Ars[1] Georgi Gaydadjiev[1] John D. Reyes[2]

[1]Delft University of Technology
Laboratory of Computer Engineering
Mekelweg 4, 2628 CD Delft, The Netherlands

[2]Intel Corporation
2200 Mission College Boulevard
Santa Clara, CA 95052, USA

E-mail: S.Hamdioui@ewi.tudelft.nl

Abstract: *This paper presents single-cell dynamic fault models for deep-submicron semiconductor memories together with their associated tests (test primitives). The test primitives are evaluated industrially, together with the traditional tests, using 65nm technology 131 Kbytes embedded SRAMs. The test results are reported, and their analysis shows the increasing importance of dynamic faults and tests, and the exceptional effectiveness of using back-to-back operations (with complementary data values) along bit lines during memory testing.*

Key words: *static faults, dynamic faults, fault primitives, fault models, memory tests, fault coverage.*

1 Introduction

The continued decrease of feature sizes in deep-submicron technology is the source of new defects and faults that strongly depend on stresses and operation sequences for their detection; issues like process variation causing threshold voltage deviation, the increasing influence of parasitics, cross talk, propagation delays, the increase in power supply noise and the reduction in the noise margin are just some examples. In this paper one of the important fault classes for deep-submicron memory technology will be addresses. Such class is called *dynamic faults* [2, 5, 9].

Dynamic faults require the application of *more than one operation sequentially* in order to be sensitized. For example, a write 1 operation followed *immediately* by a read operation causes the cell to flip to 0; however, if only a single write or a single read, or a read which does not immediately follow the write is performed, the cell will not flip. The industrial march tests have been mainly designed for static faults, and therefore may not be able to detect dynamic faults. Little has been published on dynamic faults. In [2] the existence of dynamic faults has been shown for embedded Dynamic Random Access Memories (DRAMs) based on defect injection and SPICE simulation. In [5, 9], the existence of dynamic faults for static RAMs has been proven.

This paper deals with dynamic faults. It uses a systematic way to model them, and shows the existence of other dynamic faults that have not been addressed in [2, 5, 9]. In addition, it introduces a *complete set* of dynamic fault models based on ‘two operations’ involving a *single cell*. The paper shows the shortcomings in the fault coverage of the traditional tests, and thereafter introduces new test primitives for the targeted dynamic faults. The introduced tests will be industrially evaluated together with traditional tests, and the test results will be reported.

This paper is organized as follows. Section 2 introduces the concept of *fault primitives* that will be used to classify memory faults and define the dynamic fault space in Section 3. Section 4 discusses the validation of dynamic faults for both static and dynamic RAMs. Section 5 describes the shortcoming in the fault coverage of traditional tests with respect to dynamic faults. Section 6 establishes new test primitives targeting the introduced dynamic faults. Section 7 gives the industrial evaluation and discusses the results. Section 8 ends with the conclusions.

2 Fault primitive concept and classification

In order to accurately describe the faulty behavior of memories, the concept of *Fault Primitive (FP)* [16] has been introduced. It gives a compact mathematical notation describing a single faulty behavior and is represented as $<S/F/R>$. S describes the sensitizing operation sequence that sensitizes the fault (e.g., a read ‘0’ operation from a cell containing 0 (i.e., $0r0$)), F describes the value or the behavior of the faulty cell (e.g., the cell flips from 0 to 1), while R describes the logic output level of a read operation (e.g., a wrong value 1) in case S is a read operation applied to the faulty cell. For example a read 0 destructive fault is presented as $< 0r0/1/1 >$. The concept of an FP makes it possible to give a precise definition of a *functional fault model (FFM)* as it has to be understood for memory devices [16]: *a functional fault model is a non-empty set of fault primitives.*

Let $\#O$ be defined as the number of different operations performed *sequentially* in a S. For example, if a single read operation applied to a certain cell causes that cell to flip, then $\#O = 1$. Depending on $\#O$, FPs can be divided into *static* and *dynamic* faults:

• **Static faults**: These are FPs which sensitize a fault by performing *at most* one operation; that is $\#O \leq 1$. For example, the state of the cell is always stuck at *one* ($\#O = 0$), a read operation to a certain cell causes that cell to flip ($\#O = 1$), etc. A detailed analysis of static faults together with appropriate test patterns can be found in many references like in [4, 6, 8, 12, 15].

• **Dynamic faults**: These are FPs that perform more than one operation *sequentially* in order to sensitize a fault; that is $\#O > 1$. Depending on $\#O$, a further classification can be made between *2-operation dynamic FPs* whereby $\#O = 2$, *3-operation dynamic FPs* whereby $\#O = 3$, etc. Experimental analysis [2, 5, 9] done on the new memory technologies shows that dynamic faulty behavior can take place in the absence of static faults.

3 Dynamic fault space

Dynamic faults can be divided into FPs describing single-cell faults (involving a single-cell), and FPs describing multi-cell faults (involving more than one cell). In this paper, we will restrict our analysis to single-cell faults only, because: (a) this is the first attempt to systematically analyze dynamic faults, (b) the limited space span allowed for this paper, and (c) single-cell faults are more dominant than multi-cell faults. Multi-cell FPs will be the subject of an upcoming paper.

Single-cell dynamic faults consist of FPs sensitized by applying more than one operation to a single cell *sequentially*. We will restrict our analysis to 2-operation dynamic faults because (a) they already have been shown to exist [2, 5, 9], and (b) the probability of dynamic faults decreases as the number of operations increases [3]. As mentioned in Section 2, a particular FP can be denoted as $< S/F/R >$.

S describes the *sensitizing operation sequence*, which sensitizes a fault F in the cell. Since two operations are considered, there are 18 possible Ss given below; $x, y, z \in \{0, 1\}$ and 'r' denotes a read operation and 'w' denotes a write operation.

- 8 Ss have the form '$xwywz$'; e.g., '0w1w0' denotes a write 1 operation applied to a cell whose initial state is 0; the write is followed immediately with a write 0 operation.

Table 1. List of single-cell dynamic FFMs

FFM	FPs
dRDF	$< 0r0r0/1/1 >, < 1r1r1/0/0 >,$ $< 0w0r0/1/1 >, < 0w1r1/0/0 >,$ $< 1w0r0/1/1 >, < 1w1r1/0/0 >$
dDRDF	$< 0r0r0/1/0 >, < 1r1r1/0/1 >,$ $< 0w0r0/1/0 >, < 0w1r1/0/1 >,$ $< 1w0r0/1/0 >, < 1w1r1/0/1 >$
dIRF	$< 0r0r0/0/1 >, < 1r1r1/1/0 >,$ $< 0w0r0/0/1 >, < 0w1r1/1/0 >,$ $< 1w0r0/0/1 >, < 1w1r1/1/0 >$
dTF	$< 0r0w1/0/- >, < 1r1w0/1/- >,$ $< 0w0w1/0/- >, < 1w1w0/1/- >,$ $< 1w0w1/0/- >, < 0w1w0/1/- >$
dWDF	$< 0r0w0/1/- >, < 1r1w1/0/- >,$ $< 0w0w0/1/- >, < 1w1w1/0/- >,$ $< 1w0w0/1/- >, < 0w1w1/0/- >$

- 2 Ss have the form '$xrxrx$'; e.g., '$0r0r0$' denotes two successive read 0 operations applied to a cell whose initial state is 0.
- 4 Ss have the form '$xrxwy$'; e.g., '$0r0w1$' denotes a read 0 followed immediately with write 1 applied to a cell whose initial state is 0.
- 4 Ss have the form '$xwyry$'; e.g., '$1w1r1$ denotes a write 1 followed immediately with read 1 applied to a cell whose initial state is 1.

F describes the value of the *faulty (i.e., victim)* cell (*v-cell*); $F \in \{0, 1\}$. R describes the logical value which appears at the output of the memory if the sensitizing operation applied to the v-cell is a *read* operation: $R \in \{0, 1, -\}$. A '$-$' in R means that the output data is not applicable.

Based on the values of S, F, and R, all detectable single-cell FPs can be enumerated. They consist of a total of 30 FPs. The 30 FPs are compiled into a set of 5 FFMs, and are listed in Table 1. The names of the FFMs are chosen in such a way that they represent an extension of the traditional static fault models.

1. *Dynamic Read Destructive Fault (dRDF)*: an operation (i.e., read or write) followed *immediately* by a read operation performed on a single cell changes the data in that cell, and returns an *incorrect* value on the output. The dRDF consists of six FPs; e.g., $< 0w1r1/0/0 >$: applying a 'r1' operation immediately after 'w1' operation to a cell whose initial content was 0, will cause the cell to flip to 0 and the read operation will return a wrong 0 value instead of the expected 1. The first operation involved in the sensitizing operation sequence of dRDF can be a transition write (e.g., write 1 in a cell containing 0), a non-transition write, or a read operation.

Table 2. DS fault coverage for known tests

#	Tests	Test length	dRDF	dDRDF	dIRF	dTF	dWDF	Total FC
1	SCAN [1]	4n	0/6	0/6	0/6	0/6	0/6	0/30
2	MATS+ [12]	5n	0/6	0/6	0/6	1/6	0/6	1/30
3	MATS++ [4]	6n	1/6	0/6	1/6	2/6	0/6	4/30
4	March A [13]	15n	0/6	0/6	0/6	2/6	0/6	2/30
5	March B [13]	17n	2/6	0/6	2/6	4/6	0/6	8/30
6	March C- [11, 15]	10n	0/6	0/6	0/6	2/6	0/6	2/30
7	March G [13]	23n	2/6	1/6	2/6	4/6	0/6	9/30
8	March LR [14]	14n	2/6	0/6	2/6	2/6	0/6	6/30
9	March RAW [9]	26n	6/6	4/6	6/6	2/6	2/6	20/30
10	March SS [8]	22n	4/6	0/6	4/6	2/6	2/6	12/30
11	PMOVI [7]	13n	2/6	2/6	2/6	2/6	0/6	8/30
12	Galpat [4]	6n+4nRC	0/6	0/6	0/6	2/6	0/6	2/30
13	Walking 1/0 [15]	8n+2nRC	0/6	0/6	0/6	2/6	0/6	2/30

2. *Dynamic Deceptive Read Destructive Fault (dDRDF)*: an operation followed *immediately* by a read operation performed on a single cell changes the data in that cell, and returns a *correct* value on the output. The dDRDF consists of six FPs. Here, the operation performed before the read can be a transition write, a non-transition write, or a read operation.
3. *Dynamic Incorrect Read Fault (dIRF)*: a read operation performed *immediately* after an operation (i.e., read, transition, or non-transition write) on a single cell returns an *incorrect* value on the output, while that cell remains in its correct state. The dIRF consists of six FPs.
4. *Dynamic Transition Fault (dTF)*: a transition write operation performed *immediately* after an operation (i.e., read, transition, or non-transition write) fails. The dTF consists of six FPs.
5. *Dynamic Write Destructive Fault (dWDF)*: a non-transition write operation applied *immediately* after an operation (i.e., read, transition, or non-transition write) causes that cell to flip. The dWDF consists of six FPs.

4 Validation of dynamic faults

Currently published work shows the existence of dynamic faulty behavior in the absence of the traditional static behavior. The validation of such faults for DRAMs has been shown, based on defect injection and SPICE simulation [2, 9]; e.g., the presence of an extra unwanted resistance between the bit line and the memory cell can cause the dynamic faults dRDF, dDRDF and dIRF to take a place in the absence of static faults. That means that the defect can only be detected if the dynamic analysis is considered.

Dynamic faults have also been observed in embedded SRAMs [5]. Further, the widely used 'hammer tests' (i.e., repeated read or write operations) in the industry may indicate the existence of the dynamic faults. Furthermore, the 'Holey Shmoo problem' [10] in which the L1 cache of IBM System/390 G6 microprocessor fails to pass consecutive write patterns also indicates that dynamic faults can be caused by 'a write followed immediately by another write' (i.e., dTF or dWDF). It is clear from the above, that the set of fault models for dynamic faults has to be explored, and that the appropriate test algorithms have to be established.

5 Effectiveness of the traditional tests

Table 2 summarizes the fault coverage of the most well-known memory tests; the test length of each test is also included; n denotes the size of the memory, R denotes the number of rows, and C denotes the number of columns.

In Table 2, "a/b" denotes that the test detects 'a' of the 'b' FPs of the corresponding FFM. E.g., March C- detects none of the FPs of the dRDF, while March SS detects four of them. The last column in the table (i.e., 'FC') gives the total detected FPs for the corresponding test. E.g., March RAW detects 18/30 of single-cell dynamic faults. It is clear from the table that the traditional tests written for static faults do not detect all targeted dynamic faults. This proves the need for new tests for dynamic faults.

6 Test primitives for dynamic faults

This section gives tests for the introduced single-cell dynamic faults. For each FFM, two tests will be introduced. One is written to facilitate the diagnosis of the FPs during DPM (defect per million) screening, while the other version is be optimized in terms of test length. During the industrial evaluation, all test patterns are implemented; this gives more detailed information that can be used in order to establish the importance of each FFM as well as FPs.

Table 3 Lists the tests designed for each dynamic fault. The first column gives the name of the test; e.g., Test dRDF-

Table 3. Tests for dynamic single-cell faults

Name	Test description	T.L
dRDF-Diag	$\{\Updownarrow (w0) ; \Updownarrow (w0,r0) ; \Updownarrow (r0,r0) ; \Updownarrow (w1,r1); \Updownarrow (w1,r1) ; \Updownarrow (r1,r1) ; \Updownarrow (w0,r0)\}$	$13n$
dRDF-Opt	$\{\Updownarrow (w0) ; \Updownarrow (w0,r0,r0) ; \Updownarrow (w1,r1) ; \Updownarrow (w1,r1,r1) ; \Updownarrow (w0,r0)\}$	$11n$
dDRDF-Diag	$\{\Updownarrow (w0) ; \Updownarrow (w0,r0,r0) ; \Updownarrow (r0,r0,r0) ; \Updownarrow (w1,r1,r1) ; \Updownarrow (w1,r1,r1) ; \Updownarrow (r1,r1,r1); \Updownarrow (w0,r0,r0)\}$	$19n$
dDRDF-Opt	$\{\Updownarrow (w0); \Updownarrow (w0,r0,r0,r0) ; \Updownarrow (w1,r1,r1) ; \Updownarrow (w1,r1,r1,r1) ; \Updownarrow (w0,r0,r0)\}$	$15n$
dTF-Diag	$\{\Updownarrow (w0) ; \Updownarrow (w0,w1,r1) ; \Updownarrow (r1,w0,r0) ; \Updownarrow (w1,w0,r0) ; \Updownarrow (r0,w1,r1) ; \Updownarrow (w0,w1,r1) ; \Updownarrow (w1,w0,r0)\}$	$19n$
dTF-Opt	$\{\Updownarrow (w0) ; \Updownarrow (w0,w1,r1,w0,r0) ; \Updownarrow (w1,w0,r0,w1,r1) ; \Updownarrow (w0,w1,r1) ; \Updownarrow (w1,w0,r0)\}$	$17n$
dWDF-Diag	$\{\Updownarrow (w0) ; \Updownarrow (w0,w0,r0) ; \Updownarrow (r0,w0,r0) ; \Updownarrow (w1,w1,r1) ; \Updownarrow (w1,w1,r1) ; \Updownarrow (r1,w1,r1) ; \Updownarrow (w0,w0,r0)\}$	$19n$
dWDF-Opt	$\{\Updownarrow (w0) ; \Updownarrow (w0,w0,r0,w0,r0) ; \Updownarrow (w1,w1,r1) ; \Updownarrow (w1,w1,r1,w1,r1) ; \Updownarrow (w0,w0,r0)\}$	$17n$
March DS1	$\{\Updownarrow (w0) ;\Updownarrow (w0,w0,r0,r0,r0,w0,r0) ; \Updownarrow (w1,r1,r1,w0,w0,r0,w1,r1) ;\Updownarrow (w1,w1,r1,r1,r1,w1,r1) ;$ $\Updownarrow (w0,r0,r0,w1,w1,r1,w0,r0) ; \Updownarrow (w0,w1,r1,w0,w1,r1) ; \Updownarrow (w1,w0,r0,w1,w0,r0)\}$	$43n$

Diag is the test designed for dRDF for diagnosis purpose, while Test dRDF-Opt is the optimized test for the same fault. The second column of the table gives the description of the test, while the third column gives the test length (T.L.) of the test (n denotes the memory size). The last test included in Table 3 and referred to as *March DS1* is designed to cover all single-cell dynamic faults; the test has a test length of $43n$. The reader can verify easily that each FP of each FFM is detected with its proposed test. E.g., The Test dRDF-Diag detects all dRDF FPs:

1. FP=$< 0r0r0/1/1 >$ is detected by M$_2$=$\Updownarrow$ $(r0,r0)$ (i.e., the second march element) of the test.
2. FP=$< 1r1r1/0/0 >$ is detected by M$_5$ of the test.
3. FP=$< 0w0r0/1/1 >$ is detected by M$_1$ of the test.
4. FP=$< 0w1r1/0/0 >$ is detected by M$_3$ of the test.
5. FP=$< 1w0r0/1/1 >$ is detected by M$_6$ of the test.
6. FP=$< 1w1r1/0/0 >$ is detected by M$_4$ of the test.

It should be noted that no test is included in the table for dIRF; this is because dIRF and dRDF require the same sensitizing/detection operations. Therefore dIRF can be detected with the same tests as those established for dRDF; i.e., any test detecting dRDF also detects dIRF. Outside of the memory, one cannot distinguish between the two faults since the only difference is that for dIRF the state of the cell is not changed while for dRDF it is changed. Since in this attempt of studying dynamic faults, the diagnosis of the faults will be based on the output signature, it is not possible to distinguish between the two faults.
Table 4 summarizes the introduced tests in this section, together with their fault coverage.

Table 4. Summary of single-cell dynamic tests

Test	Fault coverage					
	dRDF	dDRDF	dIRF	dTF	dWDF	Total
dRDF-Diag	6/6	3/6	6/6	0/6	0/6	15/30
dRDF-Opt	6/6	3/6	6/6	0/6	0/6	15/30
dDRDF-Diag	6/6	6/6	6/6	0/6	0/6	18/30
dDRDF-Opt	6/6	6/6	6/6	0/6	0/6	18/30
dTF-Diag	2/6	2/6	2/6	6/6	0/6	12/30
dTF-Opt	2/6	0/6	2/6	6/6	0/6	10/30
dWDF-Diag	2/6	2/6	2/6	0/6	6/6	12/30
dWDF-Opt	2/6	0/6	2/6	0/6	6/6	10/30
March DS1	6/6	6/6	6/6	6/6	6/6	30/30

7 Industrial evaluation

This section gives an industrial evaluation of the traditional tests as well as the tests designed to target single-cell dynamic faults. The memory chips considered are 65nm technology 131 Kbytes embedded SRAMs. For all tests used in this experiment (which is performed at wafer level), the same algorithmic and non-algorithmic stresses have been used.

The non-algorithmic stresses consist of the environmental conditions, externally applied to the design under test. They do not impact the sequence and/or the type of the memory operations. However, they may have a great impact on the fault coverage. The used non-algorithmic stresses in this experiment consist of: (a) high voltage (1.24V), (b) high speed (2Ghz), and (c) low temperature (-25 C).

The algorithmic stresses specify the way an algorithm is performed, and therefore they influence the sequence and/or the type of the memory operations. All tests have been implemented using the same algorithmic stresses, which consists of the address sequence 'Fast X' and the 'solid data-background'. Fast X addressing increments or decrements the address in such a way that each step goes to the next row; while solid data-background means that the data used consist of all 0s (i.e., 0000.../0000...) or all 1s.

7.1 Coverage results

All tests listed in Table 2 and in Table 3 have been implemented and applied to couple of millions of embedded SRAMs. To reduce the large data-base for analysis purposes, four classes of tests are defined and presented in Table 5.

1. Static Tests (S-Tests). They consist of four tests that mainly target static faults.
2. Diagnosis Dynamic Tests (DiagD-Tests). They consist of four dynamic tests (see Table 3) designed for diagnosis purpose to target the single-cell dynamic faults of Table 1.
3. Optimal Dynamic Tests (OptD-Tests). They consist of four optimized dynamic tests (see Table 3) designed to target the single-cell dynamic faults of Table 1.
4. Static and Dynamic Tests (SD-Tests). These are tests which were originally designed to cover static faults; however, due to their structure, they also detect some of the dynamic faults. The four tests with the most promising fault coverage for single-cell dynamic faults have been selected; see table 2.

Table 5. Classification of the tests

Static (S-Tests)	Diag. Dynamic (DiagD-Tests)	Opt. Dynamic (OptD-Tests)	Static & dynamic (SD-Tests)
Scan	dRDF-Diag	dRDF-Opt	March RAW
MATS+	dDRDF-Diag	dDRDF-Op	March SS
MATS++	dTF-Diag	dTF-Opt	March G
March C-	dWDF-Diag	dWDF-Opt	PMOVI

Figure 1 shows the venn diagrams of the failing devices for different test classes, where DiagD-Tests and OptD-Tests are compared with S-Tests and SD-Tests. The fault coverage (FC) of S-Tests is FC=335, of DiagD-Tests is FC=392, of OptD-Tests is FC=247 and that of SD-Tests is FC=484. Interestingly enough is that DiagD-Tests and OptD-Tests detect respectively 121 and 96 faults that are not detected with S-Tests. In addition, they detect respectively 18 and 13 faults that are not detected with SD-Tests. Moreover, they detect 15, respectively 11 faults that are not detected with S-Tests neither with SD-Tests. It should be noted that an analysis done on all test classes showed that the total faults detected with all test classes is 514, from which 16 faults are detected only with DiagD-Tests and/or OptD-Tests; i.e., 3.1% of the total faults are uniquely detected with DiagD-Tests and/or OptD-Tests. This clearly shows the importance of considering dynamic faults in order to achieve a high fault coverage and high product quality. Not considering dynamic faults will translate in unwanted escapes and therefore increase in DPM (Defect-per-Million) level.

7.2 Comparison of dynamic tests

Figure 2 shows a comparison of the two test classes targeting dynamic faults (DiagD-Tests and OptD-Tests) and March DS1, which is a test designed to target all single-cell dynamic faults of Table 1; see also Table 3. It is important to note that the intersection of all dynamic tests is 218 faults; these faults consist of the easy to detect static faults (e.g., stuck-at-fault) and also of single-cell dynamic faults targeted in this paper. Moreover, Figure 2 shows that DiagD-Tests have the highest FC and detect 128 faults that cannot be detected with OptD-Tests neither with March DS1; and therefore they are different than single-cell dynamic faults. This indicates that DiagD-Tests have the capability to detect other faults that are not considered in this paper (e.g., dynamic *coupling faults*, delay faults, ...).

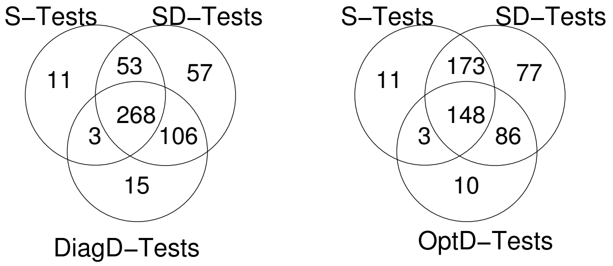

Figure 1. Venn-diagram of failing chips

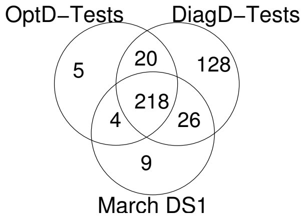

Figure 2. Comparison of the dynamic tests

Table 6 shows the *union* and the *intersections* of all dynamic tests developed in this paper; see Table 3. A die belongs to the union of two tests if at least one of the two tests detects the fault, and belongs to the intersection if *both* tests detect the faulty die. The first column in each table gives the test number; the second column the name of the test. The column ‘FC’ lists the fault coverage of the corresponding test; the column ‘UFs’ gives number of *unique faults (UFs)* each test detects. Unique faults are faults that are only detected once with a single test; e.g., March DS1 detects 9 UFs that are not detected with any other dynamic test of Table 3.

The union and the intersection of each pair of tests is shown in the rest of the table. The numbers on the diagonal give FC of the tests, which are also listed in the column ‘FC’. The part above the main diagonal shows the intersection for each test pair, while the part under the diagonal lists the union of each test pair; for example, the union of March SD1 and dTF-Diag Test is 394 while their intersection 198. Based on the table and the Venn-diagram of Figure 2, one can conclude the following:

Table 6. Intersections and unions of dynamic tests

#	Test	FC	U.F.	1	2	3	4	5	6	9	8	9
1	March DS1	257	9	**257**	166	163	149	157	198	172	119	106
2	dDRDF-Diag	178	2	269	**178**	158	150	152	131	120	116	104
3	dDRDF-Opt	173	2	267	193	**173**	149	154	131	124	113	102
4	dRDF-Diag	161	1	269	181	185	**161**	150	125	115	113	101
5	dRDF-Opt	169	0	269	195	188	180	**169**	137	124	117	109
6	dTF-Diag	335	118	394	382	377	371	367	**335**	172	94	84
7	dTF-Opt	182	2	267	240	231	228	227	345	**182**	85	78
8	dWDF-Diag	134	2	272	196	194	182	186	375	231	**134**	107
9	dWDF-Opt	115	1	266	189	186	175	175	366	239	142	**115**

- The total number of faulty chips detected with all dynamic tests is 410.
- The best test, in terms of FC, is dTF-Diag with FC=394, followed with March SD1 with FC=257.
- The best test, in terms of detecting unique faults which are only detected by a single dynamic test, is dTF-Diag with FC=394 with #UF=118.
- The best union pair in terms of the FC is 394 achieved with dTF-Diag and March DS1.

It is interesting to note that the FC achieved by dTF-Diag test is exceptionally high as compared with the other dynamic tests. Inspecting the nature and the structures of dTF-Diag Test and other dynamic tests (see Table 3) reveals that the main property that the dTF-Diag test has is that it consists of *back-to-back* operations with *complementary data values*. E.g., the second march element of dTF-Test $\Updownarrow (w0, w1, r1)$ consist of *write 0 after read 1* back-to-back. Using back-to-back operations along the bit lines (i.e, Fast X addressing) is very powerful in detecting address decoder delay faults, and dynamic/time-related faults in the peripheral circuits of the memory [17]. They are also powerful in detecting *dynamic coupling faults* since they address two different locations with successive operations.

8 conclusions

In this paper, a systematic approach to analyze dynamic faults has been described. A complete set of two-operation, single-cell dynamic faults has been developed, and appropriate tests have been introduced. The tests have been industrially evaluated together with traditional tests by applying them to advanced deep-submicron embedded SRAMs. The results showed the importance of dynamic faults and tests, and the exceptional effectiveness of using back-to-back operations during memory testing, when such operations are used with complementary data values and along the bit lines rather than the word lines. This makes the newly proposed dTF-Diag test exceptionally effective in achieving a high FC.

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] Z. Al-Ars, Ad J. van de Goor, “Static and Dynamic Behavior of Memory Cell Array Opens and Shorts in Embedded DRAMs”, *In Proc. of Design Automation and Test in Europe*, pp. 496-503, 2001.

[3] Z. Al-Ars and A.J. van de Goor, “Approximating Infinite Dynamic Behavior for DRAM Cell Defects,” *in Proc. IEEE VLSI Test Symp.*, pp. 401-406, 2002.

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

[5] S. Borri, et. al, ”Defect Oriented Dynamic Fault Models for Embedded SRAMs”, *In Proc. of European Test Symposium*, pp. 23-28, 2003.

[6] R. Dekker, *et al.,* “A Realistic Fault Model and Test Algorithms for Static Random Access Memories”, *IEEE Trans. on Computers,* **C9**(6), pp. 567-572, 1990.

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

[8] S. Hamdioui, A.J. van de Goor and M. Rodgers, “March SS: A Test for All Static Simple RAM Faults”, *In Proc. of IEEE International Workshop on Memory Technology, Design and Test*, pp. 95-100 , 2002.

[9] S. Hamdioui, Z. Al-ars and A.J. van de Goor, “Testing Static and Dynamic Faults in Random Access Memories”, *In Proc. of IEEE VLSI Test Symposium*, pp. 395-400, 2002.

[10] W. Huott, *et. al*, “The Attack of the ‘Holey Shmoos’: A Case of the Advanced DFD and Picosecond Imaging Circuit Analysis (PICA)”, *In Proc. of IEEE International Test Conference,* pp. 883-891, 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] D.S. Suk and S.M. Reddy, “A March Test for Functional Faults in Semiconductor Random-Access Memories”, *IEEE Trans. on Computers*, **C30**(12), pp. 982-985, 1981.

[14] A.J. van de Goor and G. Gaydadjiev, “March LR: A Memory Test for Realistic Linked Faults”, *In Proc. IEEE VLSI Test Symposium,* pp. 272-280, 1996.

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

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

[17] A.J. van de Goor and S. Hamdioui, ”Detecting Faults in Peripheral Circuits and an Evaluation of SRAM Tests”, *In proc. of IEEE International Test Conference*, pp. 114-134, USA, October, 2004