March Test BDN: A new March Test for Dynamic Faults

Alberto Bosio, Giorgio Di Natale

March Test BDN: A new March Test for Dynamic Faults

Alberto BOSIO, Giorgio DI NATALE
Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier
Université Montpellier II / CNRS UMR 5506
161 rue Ada, 34392 Montpellier Cedex 5, France
{bosio,dinatale}@lirmm.fr

***Abstract*- High-density components and process scaling lead more and more to the occurrence of new class of dynamic faults, especially in Static Random Access Memories (SRAMs), thus requiring more and more sophisticated test algorithms. Among the different types of algorithms proposed for testing SRAMs, March Tests have proven to be the most performing due to their low complexity, their simplicity and regular structure. Several March Tests for dynamic faults have been published, with different fault coverage. In this paper we propose March BDN, an extended version of the March AB. We will prove that it is able to increase the fault coverage in order to target latest dynamic faults. We show that the proposed March BDN has the highest known fault coverage compared to March Tests with the same complexity.**

I. INTRODUCTION

Memories are one of the most important components in digital systems, and semiconductor memories are nowadays one of the fastest growing technologies. Currently, *System-On-Chip* (SOC) technology allows to embed in a single chip all the components and functions that historically were placed on a hardware board. Within SOCs, embedded memories are the densest components, accounting for up to 90% of chips area [1]. It is thus common to find on a single chip tens of memories of different types, sizes, access protocols and timing. Moreover, they can recursively be embedded in embedded cores. The high density of their cells array makes memories extremely vulnerable to physical defects.

In the last years, the so called *Static Faults* (e.g., stuck-at faults, coupling faults, ...) [2] have been the predominant fault type. They are characterized by being sensitized by the execution of just a single memory operation. New faulty behaviors occur in latest technologies [3] [4]. As an example, a write operation on a memory cell, immediately followed by a read operation, may cause the cell to flip. On the contrary, if just a single write or a single read, or a read that does not immediately follow a write are performed, the cell does not flip. These behaviors cannot be modeled as Static Faults since they require more than one operation to be sensitized, and are referred to as *Dynamic Faults*. The set of possible Dynamic Faults is theoretically unlimited and whenever a new fault is observed a new custom test algorithm has to be designed.

Although ad-hoc testing strategies are needed to address the peculiar set of faults that can affects SRAMs, their regular structure allows adopting particularly simple algorithms, the most popular one being *March Tests* [2]. While several March Tests targeting Static Faults have been proposed [6][5][7], few March Tests have been developed to detect Dynamic Faults. [8] presents March RAW1 and RAW, of length 13*n* and 26*n*, respectively; the former one covers single-cell dynamic faults whereas the latter one detects two-cell dynamic faults (see Section 2). In [9], we presented March AB (22*n*), able to cover the whole set of dynamic fault models presented in [10]. Moreover, March AB is also able to cover both static linked and un-linked faults.

In [11][12][13][14][15][16] authors show that in newer technologies the set of dynamic faults is further incremented w.r.t. the taxonomy proposed in [10]. They also give the test solution able to deal with the new class of dynamic faults. The main drawback of the proposed solution is that each fault model is covered by a specific march test. Thus, to cover the whole set of dynamic fault model several tests must be applied. Therefore the overall test time dramatically increases.

In this paper we introduce the March Test BDN that is extended modified version of March AB [9]. The newer version is able to improve the fault coverage of March AB. In particular, it also detects new classes of dynamic faults, while keeping the same complexity of March AB.

To better identify the new set of target dynamic faults, we present them resorting to the Fault Primitive formalism introduced in [14].

To analytically prove the correctness and the efficiency of the proposed March Tests, we defined the coverage conditions for each fault model, i.e. the sequence of memory operations needed to sensitize and detect the fault effects. We thus prove that March BDN respects the coverage conditions for each fault in the fault list. Furthermore, we compare its fault coverage with respect to already published algorithms. The correctness of the proposed March Test has been also proved by fault simulation experiments using a memory fault simulator [15].

The paper is structured as follows: Section 2 introduces the Fault Primitive formalism and the dynamic faults classification; Section 3 presents the new March Tests. Section 4 details the complete list of the fault coverage conditions. Comparison evaluations are reported in Section 5, while Section 6 summarizes the main contributions and outlines future research activities.

1-4244-2577-8/08/$20.00 ©2008 IEEE

II. FAULT MODEL & TAXONOMY

For test purposes, faults in memories are usually modeled as Functional Faults. A *Functional Fault Model* (FFM) is a deviation of the memory behavior from the expected one under a set of performed operations. A FFM involves one or more *Faulty Memory Cells* (FMC) classified in two categories: *Aggressor cells* (*a*-cells), i.e., the memory cells that sensitize a given FFM and *Victim cells* (*v*-cells), i.e., the memory cells that show the effect of a FFM. Each FFM can be described by a set of Fault Primitives (FPs) [10]. A Fault Primitive is identified by <S/F/R>, it represents the difference between an expected (good), and the observed (faulty) memory behavior in which:

- *S = Sa ; [Sv]* is a sequence of *m* operations and/or conditions, respectively applied to *a*-cell (*Sa*)and *v*-cell (*Sv*), needed to sensitize the given fault. The *j*-th operation is represented as $op_j = c(iOd)_j$ where *c* is the cell address; $i \in \{0,1\}$ is the initial value stored in a memory cell; $O \in \{w,r\}$ is the type of operation performed on a cell; $d \in \{0,1\}$ in case of write operation represents the data to be written into memory *cell*. *Sv* is omitted when the FP correspond to a single cell memory fault, because it involves just one cell
- *F* is the faulty behavior, i.e., the value (state) stored in the victim cells after applying *S*
- *R* is the sequence of values read on the aggressor cell when applying *S*.

As an example FP = <$0w_1$; 0/1/- > means that the operation 'w1' performed on the *a*-cell, when the initial state is 0 for both *a* and *v* cells, causes the *v*-cell to flip.

The considered dynamic fault models can appear both in the cell array, in the R/W circuitry and in the address decoder circuitry [5]. In the following sub sections we provide the taxonomy of the newer dynamic faults.

A. Dynamic fault in the array cell

Here we give the taxonomy of the newer dynamic faults affecting the memory array cell, such dynamic faults are:

- *Dynamic Read Destructive Fault (dRDF)* [10] [14]: where a write operation immediately followed by multiple read cooperation changes the logical value stored in the memory cell and returns an incorrect output;
- *Dynamic Data Retention Fault (dDRF)* [2]: occurs when a memory cell loses its previously stored logic value after a certain period of time during which it has not been accessed. This faults can be sensitized by a specific sequence of read operations as detailed in [15];
- *Leakage Read Fault (LRF)* [11]: When in a memory column most of the cells store the same value $x \in \{0,1\}$, the leakage currents, through the pass transistors of the unselected cells may affect the read operations of cells storing the value $y = \text{not}(x)$ is expected whereas *x* is read.

Table 1 summarizes the above fault models in term of FPs. The first column reports the Functional Fault Model, while the second column details the description of the FFM in terms of FPs. The notation r_0^n (r_1^n) means a read operations sequence of length equal to *n* ($n \geq 1$). For the LRF, *k* corresponds to the number of columns in the memory under test. Note that the FPs of LRF maximize the sensitization condition since all the cell in the column are set to the value 0 (1) except the faulty one.

TABLE I
DYNAMIC FAULTS IN THE ARRAY CELL

FFM	*Fault Primitives*
dRDF	<$0w_0\ r_0^n$ /1/1>,<$1w_1r_1^n$ /0/0>, <$0w_1r_1^n$ /0/0>,<$1w_0r_0^n$ /1/1>
dDRF	<$0r_0^n$ /1/1>,<$1r_1^n$ /0/0>,
LRF	<0(0)1(0) k-2(0)k-1($1r_1$)/0/0>, <0(1)1(1) k-2(1)k-1($0r_0$)/1/1>

B. Dynamic fault in the address decoder

Here we give the taxonomy of the newer dynamic fault affecting the address decoder. Such dynamic faults are:

- *Address Decoder Open Fault (ADOF) and Resistive-ADOF* [17]: A decoder is said to have an ADOF/R-ADOF if changing only one bit on its address results in selecting this new address but also the previous one. Consequently, two core-cells are selected at the same time for a read or a write operation.

TABLE II
DYNAMIC FAULTS IN THE ADDRESS DECODER

FFM	*Fault Primitives*
ADOF	<w_0 ; w_1/0/->,<w_1 ; w_0/1/->
R-ADOF	<w_0 ; w_1/0/->,<w_1 ; w_0/1/->

Table 2 summarizes the above fault models in term of FPs. The first column reports the Functional Fault Model, while the second column details the description of the FFM in terms of FPs. Note that the two fault models have the same FPs, in fact the difference between the ADOF and R-ADOF is that the R-ADOF is due to the resistive-open defects in the address decoder. It is also important to mention that the difference between the addresses of the aggressor (a) and the victim (v) must be one bit. This constraint is formalized by using the hamming distance Hd (a,v) = 1.

C. Dynamic fault in the R/W circuitry

Here we give the taxonomy of the newly introduced dynamic faults affecting the R/W circuitry [14] [15]. In particular, we consider faults affecting the sense amplifier and the write driver. Such dynamic faults are:

- *Dynamic Two-Cell Incorrect Read Fault (d2cIRF)* [13]: a sense amplifier is said to have a d2cIRF if it is not able to read any value. So, the read data value at the output is the one previously stored in the data output circuitry;

- *Slow Write Driver Fault (SWDF)* [12]: a write driver is said to have a SWDF if it cannot act a w0 (w1) when this operation is preceded by a w1 (w0). This results in a core-cell that does not change its data content.

TABLE III
DYNAMIC FAULTS IN THE R/W CIRCUITRY

FFM	*Fault Primitives*
d2cIRF	$<0r_0 ; 1r_1/1/0>$, $<1r_1 ; 0r_0/0/1>$
SWDF	$<1w_0w_1/0/->$, $<0w_1w_0/1/->$, $<0w_0w_1/0/->$, $<1w_1w_0/1/->$

Table 3 summarizes the above fault models in term of FPs. The first column reports the Functional Fault Model, while the second column details the description of the FFM in terms of FPs.

III. MARCH TEST BDN

A *March Test* is a test algorithm composed of a sequence of *March Elements* [2]. Each *March Element* (ME) is a sequence of memory operations applied sequentially on a certain memory cell before proceeding to the next one. The way in which one moves from a certain address to another one is called *Address Order* (AO). The AO characterizes the ME. Hereinafter, we shall denote a March Test using a '{…}' bracket and a March Element using a '(…)' bracket. The i-th operation is defined as op_i where $op_i \in \{w_d, r_d\}$, $d \in \{0,1\}$ in which 'r_d' means "read the content of the memory cell and verify that its value is equal to d ".

The complexity of a March Test is in the order of $O(c \cdot n)$, being c the overall number memory operations composing the different march elements and n the number of memory cells. In other words, a given March Test is linear w.r.t. the size of the memory under test.

Figure 1 shows the **March BDN**, with a complexity of $22n$.

$$\{\Uparrow (w_a)\ \Downarrow(r_a,w_b,r_b,w_b,r_b)\ \Downarrow(r_b,w_a,r_a,w_a,r_a)\ \Uparrow(r_a,w_b,r_b,w_b,r_b)\ \Uparrow(r_b,w_a,r_a,w_a,r_a)\ \Uparrow(r_a)\}$$
$$M1 \quad M2 \quad M3 \quad M4 \quad M5 \quad M6$$

Figure 1. March BDN $O(n) = 22n$

March BDN is an extension of March AB [9]. In particular we resort to the degrees of freedom of a March test [18] to derive March BDN from March AB. The exploited degrees of freedom are:

1. The address sequences can be freely chosen as long as all addresses occur exactly once and the sequence is reversible;
2. The data within a read/write operation have not to be necessarily equivalent for all memory addresses, as long as the detection probabilities for basic faults are not affected.

Thanks to the first degree of freedom, we can modify the address order of march elements. Basically we propose an address order that follows line after line, respecting the constraint that each consecutive address has a hamming distance equal to one. The second degree is exploited to implement a logical alternated data sequence in each ME. In particular, $a = \text{not}(b)$ and $b \in \{0,1\}$.

The first consideration is that March BDN still has the same coverage of March AB, since [18] proves that the degree of freedom does not modify the fault coverage of a March test. Thus, March BDN is able to detect the whole set of realistic static linked and un-linked faults as well as the dynamic faults presented in [10]. Moreover, thanks to the modification, March BDN improves the fault coverage by addressing dynamic faults introduced in Section 2.

In the following section, we define a set of *Fault Coverage Conditions* (FCC) needed to detect the target faults of Section 2. Each FCC specifies the March Elements able to detect the target fault, and we will prove that March BDN satisfies all those conditions.

III. . FAULT COVERAGE CONDITIONS

A *Fault Coverage Conditions* (FCC) represents a MEs sequence, formalized with the March Test notation (Section 3). It can be automatically derived from the FP formalism in order to determine the set of March tests targeting each FPs.

In order to ensure the correctness of March BDN we verify that it includes the occurrence of the FCCs. Next sections detail the FCCs and prove their coverage.

A. FCC targeting dynamic faults in the array cell

The dRDF requires the sensitization sequence $w_x\ r_x^n$ where n is the number of required read operations. In [19] the authors shown that any memory operations performed on a memory cell i cause a so called Read Equivalent Stress (RES) on all the memory cells on the same line of the cell i. The RES, is equivalent to perform a read operation on the memory cell. Exploiting this principle, FCC1 is able to detect the dRDF.

$$\textbf{FCC1: } \Downarrow(w_x,r_x)\ \Downarrow(w_y,r_y)\ \Uparrow(w_x,r_x)\ \Uparrow(w_y,r_y)$$

Where $x = \text{not}(y)$ and $y \in \{0,1\}$. Moreover, the address order is line after line in order to ensure the detection of dRDF as shown in [19]. Since the dDRF requires the sensitizations sequence r_x^n , FCC1 is also able to detect the dDRF.

$$\textbf{FCC2: } \Updownarrow(w_y)\ \Updownarrow(,w_x,r_x)\ \Updownarrow(w_y,r_y)$$

The FCC2 is able to cover the LRF. In fact the first ME set the memory to value y ($y \in \{0,1\}$), then the second ME apply w_x,r_x ,$x = \text{not}(y)$, on each cell, that ensure the sensitization of the LRF as expressed in Table 1 and shown in [11]. The address order is not important therefore we can adopt the line after line

B. FCC targeting dynamic faults in the address decoder

The ADOF requires the sensitization sequence $w_x\, w_y$ where x = not(y) and $y \in \{0,1\}$ and the address of the two memories cell has a hamming distance equal to 1.

FCC3: $\Updownarrow(w_a)\ \Downarrow(r_a, w_b)\ \Uparrow(r_b, w_a)\ \Updownarrow(r_a)$

FCC3 is able to cover ADOF, since it includes the sensitization sequence. To guarantee the coverage of the ADOF two constraints have to be satisfied, (i) the address order of each ME must be able to generate the sequence of address having the hamming distance equal to 1. (ii) the data sequence implement a logic alternate value in each ME. In this case a = not(b) and $b \in \{0,1\}$. These two constraints are proven in [16].

FCC3 is also able to detect the R-ADOF since it is modeled with the same FPs of ADOF.

C. FCC targeting dynamic faults in the R/W circuitry

The SWDF is detected by the FCC4 where x = not(y) and $y \in \{0,1\}$.

FCC4: $\Updownarrow(w_y)\ \Downarrow(w_x, r_x)\ \Downarrow(w_y, r_y)\ \Uparrow(w_x, r_x)\ \Uparrow(w_y, r_y)\ \Updownarrow(r_y)$

The FCC4 guarantees the execution of the sequence of operations w_0w_1 (or w_1w_0) required to sensitize the fault [12]. Finally, the d2cIRF is also covered by the FCC4 as proven in [13].

V. March BDN Validation

In order to validate March BDN we have to prove that it includes the FCCs introduced in Section 4. First of all it is evident that FCC4 includes FCC3, FCC2 and FCC1 in terms of operations. We have only to specify the address order that is line after line with a hamming distance equal to 1 and he logic alternate data sequence in order to ensure that all the faults are covered. Table 4 presents the obtained FCC4.

TABLE IV
FCC4

#	March Elements
M0	$\Uparrow(w_a)$
M1	$\Downarrow(r_a, w_b)$
M2	$\Downarrow(r_b, w_a)$
M3	$\Uparrow(r_a, w_b)$
M4	$\Uparrow(r_b, w_a)$
M5	$\Uparrow(r_a)$

Now it is easy to see the each ME specified in Table 4 is included in March BDN with the same addressing order and the alternate logic value.

VI. March Test Comparison

We compared our test with the following set of March tests composing the state of the art in terms of fault coverage:

- March AB [9]: 22*n,* able to cover dynamic un-linked faults, static linked and un-linked;
- March RAW [8]: 26*n,* able to cover dynamic un-linked faults and static un-linked;
- March C- [2]: 10*n,* able to cover static un-linked faults;
- March iC- [16]: 10*n,* able to cover the presented set of dynamic un-linked faults;

Each March algorithm has been simulated using an extended version of the memory fault simulator presented in [20]. Modifications have been done in order to deal with the degrees of freedom of a March test [18].

Table 5 summarizes the simulation results in terms of fault percentage covered by each March Test and its complexity (O(n)). It targets the set of dynamic faults presented in [10] and the ones shown in this paper (column presented paper). Comparison results show that the proposed March Test provides the same fault coverage of the best known one, without increasing the complexity.

It is important to note that without March BDN the test of the new set of dynamic faults is possible only by applying at least two March Tests: March AB and March iC-. Therefore, the final complexity would be equal to 32*n*. Since March BDN (O(22n)) provides the same coverage of the sum of the two March Test (O(32n)), we reduce the test complexity, and therefore the test time, of a significant percentage of 31%.

TABLE V
MARCH TEST COMPARISON

MT	O(n)	Dynamic Faults	Presented Faults
C-	10*n*	0.87%	0%
iC-	10*n*	14%	100%
AB	22*n*	100%	77%
BDN	22*n*	100%	100%
RAW	26*n*	100%	76%

VII. Conclusion

This paper proposed March BDN, a new march test targeting the latest introduced dynamic faults. A detailed description of the coverage conditions needed to detect each fault has been proposed, and the correctness of Marc BDN has been proved by demonstrating that it satisfies all the coverage conditions.

Moreover we compared March BDN with state-of-the-art algorithms resorting to fault simulation experiments, showing that our test provides the maximum coverage without increasing the test time. March BDN allows having a single march test addressing an extended set of faults. Furthermore, due to its regular and symmetric structure, March BDN

becomes a natural candidate for memory BIST architectures, making our solution very attractive for the industry.

REFERENCES

[1] International Technology Roadmap for Semiconductors, *"International technology roadmap for semiconductors 2004 Update"*, http://public.itrs.net/Home.htm, 2004.

[2] A. J. van de Goor, *"Testing Semiconductor Memories: theory and practice"*, Wiley, Chichester (UK), 1991.

[3] Z. Al-Ars, Ad J. van de Goor, "*Static and Dynamic Behavior of Memory Cell Array Opens and Shorts in Embedded DRAMs*", DATE 2001, IEEE Design Automation and Test in Europe, 2001, pp. 496-503.

[4] S. Hamdioui, R. Wadsworth, J.D. Reyes, A. J. van de Goor, "Importance of Dynamic Faults for new SRAM Technologies", *ETW 2003, 8th IEEE European Test Workshop*, 2003, pp. 29 -34.

[5] R. Dekker, F. Beenker, L. Thijssen, *"*A Realistic Fault Model and Test Algorithms for Satic Random Acces Memory*"*, IEEE Transaction on Computer-Aided Design, Volume: 9, Issue: 6, June 1990.

[6] R.D. Adams and E.S. Cooley, *"Analysis of a Deceptive Destructive Read Memory fault Model and Recommended Testing",* NATW 1996. 5th IEEE North Atlantic Test Workshop, 1996.

[7] D. S. Suk, S. M. Reddy, *"A March Test for Functional Faults in Semiconductor Random-Access Memory"* IEEE Transaction on Computer-Aided Design, Volume: 30, Issue: 12, 1981.

[8] S. Hamdioui, Z. Al-Ars, A. J. van de Goor, "Testing Static and Dynamic Faults in Random Access Memories", *VTS 2002, 20th IEEE VLSI Test Symposium*, 2002, pp.

[9] A. Benso, A. Bosio, S. Di Carlo, G. Di Natale, P. Prinetto, "*March AB, March AB1: New March Tests for Unlinked Dynamic Memory Faults*", ITC 2005, IEEE International Test Conference, 2005.

[10] A. J. van de Goor, Z. Al-Ars, *"Functional Memory Faults: A Formal Notation and a Taxonomy"*, VTS 2000, 18th IEEE VLSI Test Symposium, 2000, pp. 281-289.

[11] L. Dilillo, B. Al-Hashimi, P. Rosinger, P. Girard, "*Leakage Read Fault in Nanoscale SRAM: Analysis, Test and Diagnosis*", IEEE International Design and Test Workshop, November 19-20, 2006.

[12] A. Ney, P. Girard, C. Landrault, S. Pravossoudovitch, A. Virazel and M. Bastian, "Slow Write Driver Faults in 65nm Technology SRAM: Analysis and March Test Solution", Proc. of Design Automation and Test in Europe, 2007.

[13] A. Ney, P. Girard, C. Landrault, S. Pravossoudovitch, A. Virazel and M. Bastian, "Dynamic Two-Cell Incorrect Read Fault due to Resistive-Open Defects in the Sense Amplifiers of SRAMs", Proc. of European Test Symposium, pp 97-102, 2007.

[14] S. Borri, M. Hage Hassan, P. Girard, C. Landrault, S. Pravossoudovitch and A. Virazel, "*Defect-Oriented Dynamic Fault Models for Embedded SRAMs*", Proc. of European Test Workshop, pp. 23-27, 2003.

[15] L. Dilillo, P. Girard, S. Pravossoudovitch, A. Virazel and M. Bastian Hage-Hassan, "Data Retention Fault in SRAM Memories: Analysis and Detection Procedures", Proc. of VLSI Test Symposium, pp 183-188, 2005.

[16] L. Dilillo, P. Girard, S. Pravossoudovitch, A. Virazel and S. Borri, "*March iC-: An Improved Version of March C- for ADOFs Detection*", VLSI Test Symposium, pp. 129-134, 2004.

[17] M. Sachdev, "Open Defects in CMOS RAM Address Decoders", IEEE Design & Test of Computers, Vol.14, N°2, Apr-Jun 1997, pp. 26-33.

[18] D. Niggemeyer, M. Redeker and J. Otterstedt, "Integration of Non-classical Faults in Standard March Tests", Records of the IEEE Int. Workshop on Memory Tech., Design and Testing, pp. 91-96, 1998.

[19] L. Dilillo, P. Girard, S. Pravossoudovitch, A. Virazel, S. Borr and M. Hage Hassan, "Dynamic Read Destructive Fault in Embedded-SRAMs: Analysis and March Test Solution", Proc. of European Test Symposium, pp 140-145, 2004.

[20] A. Benso, S. Di Carlo, G. Di Natale, P. Prinetto, *"*S*pecification and design of a new memory fault simulator"*, ATS 2002, 11th IEEE Asian Test Symposium, 2002.pp. 92 – 97.