March Test for Static 3-Coupling Faults in Random-Access Memories

PETRU CAŞCAVAL
Dept. of Computer Science and Engineering
"Gh. Asachi" Technical University of Iaşi
53A, D. Mangeron, 700050, Iaşi
ROMANIA

Abstract: – A march test of length $30n$ for detecting static 3-coupling faults in $n \times 1$ random-access memories (RAMs) is presented. To reduce the length of the test, only the coupling faults between physically adjacent memory cells have been considered. The test assumes that the storage cells are arranged in a rectangular grid and that the mapping from logical addresses to physical cell locations is known completely. In this paper any memory fault is modelled by a set of primitive faults. The ability of this march test to detect all primitive 3-coupling faults is proved by using an Eulerian graph model. To compare this march test with other published tests, simulation results are also presented in this paper.

Key–Words: RAM testing, Static 3-coupling faults, March test, Memory fault simulation.

1 Introduction

Memory test procedures are constrained by the following conflicting requirements:
a) to detect a wide variety of memory faults;
b) to reduce the number of memory operations in order to allow large memories to be tested in an acceptable period of time.

Rapid developments in semiconductor technology have resulted in continuing growth of larger and denser random-access memories on a single chip (now 256 Mb and more). With increasing densities, certain types of faults, such as *v*-coupling faults or dynamic faults, which are harder-to-detect, are becoming more important. In addition, more time is required to test memories because of their increasing size thus it is necessary to identify more efficient tests, with the ability to detect complex faults, tests that require test time on the order of *n.*

In this work we focus on static 3-coupling faults and propose an efficient march test. This march test is an improvement on the results presented in [1] and [2], where similar models are considered and march tests of length $38n$ and $34n$, respectively, are given.

For the 3-coupling fault model, a memory test that requires $n+32n\log_2 n$ operations is given by Nair, Thatte, and Abraham [3] (Algorithm B, *NTA*(*B*) in this article). In [4] a new test of length $n+24n\log_2 n$ is proposed by Papachristou and Sahgal (*PS*(*B*) in this article). Two more efficient memory tests, *S3CTEST* and *S3CTEST2*, are proposed by Cockburn in [5]. Taking into account the fault model considered in this paper and based on the specification given by Cockburn, we consider the Cockburn tests with the sequence of operations $(r_u w_{\bar{u}} r_{\bar{u}} w_u)$ instead of the reduced sequence $(r_u w_{\bar{u}} w_u)$. Consequently, *S3CTEST* and *S3CTEST2* are tests of approximate length $5n\log_2 n+22.5n$ and $5n\log_2 n +5n[\log_2(1+\log_2 n)]+11n$, respectively. For the memory chips currently available, these tests take a long time to perform. For example, to test a 64 Mb memory chip assuming a cycle time of 60 ns, *PS*(*B*) and *S3CTEST* take about 44min 24s and 10 min 54s, respectively.

These memory tests are long because the authors have assumed that the coupled cells can be anywhere in the memory.

To reduce the length of the test, we limit ourselves to the coupling faults that affect only physically adjacent memory cells. For this model, a march test that needs only $30n$ operations is proposed. This test assumes that the mapping from logical addresses to physical cell locations is known completely.

The remainder of this paper is organised as follows. Section 2 defines the fault model and Section 3 introduces notations and preliminary considerations. Section 4 presents the new march test and analyses the ability of this test to detect all primitive 3-coupling faults. To compare the new march test with other published tests, simulation results are presented in Section 5.

2 Memory Fault Model

This paper treats the problem of coupling faults in random-access memories. Because the address decoders, sense amplifiers and write drivers are easier to test, we assume that these modules are

fault-free and we focus only on the static faults in the memory cell array where difficult-to-detect faults may exist.

Many different faults can occur in a memory cell array. These can be classified as faults which involve only a single cell (such as stuck-at, stuck-open, transition and data retention faults) and faults where a cell or set of cells influences the behaviour of another cell (such as coupling faults and pattern sensitive faults) [6].

Coupling faults involve v cells ($v \geq 2$). In a set of coupled cells an active and/or a passive influence on a victim cell may exist [6]. Accordingly, coupling faults can be divided into transition and state coupling faults.

1) *Transition coupling faults* (*TCFs*). Consider a set of v–coupled cells, $v \geq 2$. The transition coupling fault is used to represent the situation when write operations addressed to one memory cell of the set, say cell j, cause the state of another cell in the set, say cell i, to change from 0 to 1 or from 1 to 0, while the v-2 remaining cells hold a specific pattern. Cell i is called the coupled (victim) cell and cell j is called the coupling (aggressor) cell. If $v>2$ then the v-2 remaining cells are called the enabling cells. We write down this fault as $j \rightarrow i$ TCF. For the case in which two cells are coupled (v=2), two types of TCFs are usually considered: inversion coupling fault (CFin) and idempotent coupling fault (CFid) as defined in [6].

2) *State coupling faults* (*SCFs*). The state coupling fault is used to represent the situation when a cell in a set of v-coupled cells ($v \geq 2$), say cell i, fails to undergo a $0 \rightarrow 1$ or a $1 \rightarrow 0$ transition when the complement of the contents of the memory cell is written into the cell, while the remaining v-1 cells in the set (enabling cells) hold a specific pattern. In such a case, we say that the enabling cells have a passive influence on the coupled (victim) cell and call this fault an i-state coupling fault (i-SCF).

In this work we have limited ourselves to the model with at most three coupled cells and we assume that only physically adjacent cells can be 3–coupled. For a set of 3–coupled cells, six patterns P_1, P_2, P_3, P_4, P_5 and P_6 are accepted, as shown in Fig. 1. We call these models of 3-coupling faults, which comprise only physically adjacent cells, reduced 3-coupling.

Remark 1. Because in our model the 3-coupled cells are physically adjacent, the transition coupling faults and the state coupling faults can also be considered as active neighbourhood pattern sensitive faults (ANPSFs) and passive neighbourhood pattern sensitive faults (PNPSFs), respectively [7]. But in our model any cell in a set of v-coupled cells can be a victim cell, not only the central base cell.

In the memory, one or more sets of coupled cells may exist. As in [3], [4] and [5] we assume that the pairs of sets of coupled cells are disjoint.

Definition 1. A *triggering transition* is defined as one that is initiated by the testing algorithm by writing into a cell the complement of the previous logic value of the cell.

We assume that a memory fault can be activated only by a triggering transition into a cell. We do not consider in this work disturb coupling faults which can be activated by either read or write operations (as is possible for dynamic coupling faults).

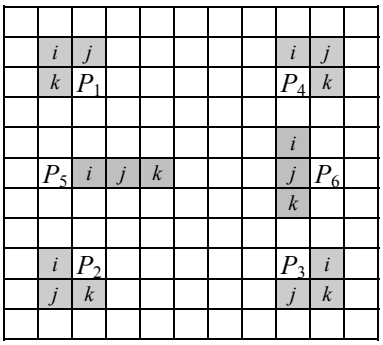

Fig. 1 – Patterns for three physically adjacent cells.

3 Notations and Preliminaries

We use the following notations to describe operations on RAMs :

- $x \in \{0,1\}$ denotes that a cell is in a logical state x;
- r (r^i) – the read operation on a cell (cell i).
- w_x (w_x^i) – the operation of writing x into a cell (cell i), $x \in \{0,1\}$.
- w_c (w_c^i) – the operation of writing the complement of the previous state of a cell (cell i).
- $\uparrow(\uparrow i)$ – the operation of writing 1 into a cell (cell i) when the previous state of the cell was 0.
- $\downarrow(\downarrow i)$ – the operation of writing 0 into a cell (cell i) when the previous state of the cell was 1.

Consider a set S of v–coupled cells. A state of set S is given by the logical state of each cell in S. In order to describe a failed write operation in S, we use a vector F with $2v$ elements grouped in two parts, separated by ":". The first part shows the initial state of set S and the triggering transition which activates the fault, whereas the second part shows the state of set S after the triggering transition is carried out.

Vector F is composed by the symbols 0, 1, ↓ and ↑. Only one symbol in vector F can be ↑ or ↓. For example, for a set of cells $S=\{i,j,k\}$,

- vector F_1 = <0,↑,0:0,0,0> describes a SCF in which the triggering transition ↑j fails to write 1 into cell j while cells i and k are in the state 0, and,
- vector F_2 = <↑,0,0:1,0,1> describes a TCF in which the triggering transition ↑i changes the state of cell k from 0 to 1 if cell j is in the state 0.

To emphasis the state of cell affected by the fault (the victim cell) we use a logic variable D (the well-known Roth's notation), as follows:

$$D = \begin{cases} 1 - \text{the cell is fault free} \\ 0 - \text{the coupling fault has been activated} \end{cases}$$

Thus, the faults previously defined become:

F_1 = <0,↑,0:0,D,0> and F_2 = <↑,0,0:1,0, $\overline{D}$ >,

where $\overline{D}$ = NOT D.

Definition 2. A memory fault that affects a cell in a set of cells S is called a *primitive fault* if and only if it is activated by a single cell transition in set S. Note that, only one vector F is necessary to describe a primitive fault. If at least two vectors are necessary for a complete description, the fault is called a *complex fault.*

Any complex fault can be modelled as a set of distinct primitive faults simultaneously present in the memory. Table 1 presents all primitive 3-coupling faults which may affect cell i in a set of cells $S=\{i,j,k\}$. As shown in Table 1, for each victim cell in a set S of v–coupled cells, $v \cdot 2^v$ primitive faults must be considered.

Definition 3. An *interacting fault* denotes a complex fault comprising two primitive faults with contrary effects on the same victim cell. Thus, an interacting fault is described by two vectors F_1 and F_2 which contain in the same cell position the symbols D and $\overline{D}$, respectively, or vice-versa.

Table 1. Primitive 3-coupling faults which affect cell i in set of cells $S=\{i,j,k\}$.

Vector F i–SCFs	Vector F $j\rightarrow i$ TCFs	Vector F $k\rightarrow i$ TCFs
<↑,0,0 : D,0,0>	<0,↑,0 : $\overline{D}$,1,0>	<0,0,↑ : $\overline{D}$,0,1>
<↑,0,1: D,0,1>	<0,↑,1 : $\overline{D}$,1,1>	<0,1,↑ : $\overline{D}$,1,1>
<↑,1,0 : D,1,0>	<1,↑,0 : D,1,0>	<1,0,↑ : D,0,1>
<↑,1,1 : D,1,1>	<1,↑,1 : D,1,1>	<1,1,↑ : D,1,1>
<↓,0,0 : $\overline{D}$,0,0>	<0,↓,0 : $\overline{D}$,0,0>	<0,0,↓ : $\overline{D}$,0,0>
<↓,0,1 : $\overline{D}$,0,1>	<0,↓,1 : $\overline{D}$,0,1>	<0,1,↓ : $\overline{D}$,1,0>
<↓,1,0 : $\overline{D}$,1,0>	<1,↓,0 : D,0,0>	<1,0,↓ : D,0,0>
<↓,1,1 : $\overline{D}$,1,1>	<1,↓,1 : D,0,1>	<1,1,↓ : D,1,0>

Examples of complex fault modelling are:

- Take 2-coupled cells with cell i the coupling cell and cell j the coupled cell. The transition ↑i changes the state of cell j from 0 to 1 and the transition ↓i changes the state of cell j from 1 to 0. This interacting linked fault can be modelled by two vectors,

  F_1 = <↑,0:1, $\overline{D}$ > and F_2 = <↓,1:0,D>.
- Consider a cell j that is a victim cell of two aggressor cells i and k. When cells i, j and k are in the state 0, the transition ↑i or ↑k changes the state of cell j from 0 to 1. This non-interacting linked fault can be modelled by two vectors,

  F_1 = <↑,0,0:1, $\overline{D}$,0> and F_2 = <0,0,↑:0, $\overline{D}$,1>.

To test and find a fault in a memory we need to be able to:

a) activate the fault by a proper triggering transition, and,
b) observe the fault by reading the changed value of the cell affected by the fault.

Proposition 1. Assume that in a set of cells at most one primitive fault may exist. The next three conditions are necessary and sufficient for a test to detect any primitive fault that affects a cell in a set S of coupled cells:

- *Condition 1*. The test must force all the possible cell transitions in the set of coupled cells in order to activate any fault.
- *Condition 2*. After a triggering transition into a cell in set S, the test must read the cell to check if the state has changed before another triggering transition into the cell is allowed to occur. This condition is required to detect any primitive SCF activated in set S.
- *Condition 3*. For each possible coupled cell c in S, after one or more triggering transition in other cells in the set, the test must read cell c, prior to a triggering transition into cell c, to check if the state has been changed by a triggering transition in other possible coupling cell. This condition is required to detect any primitive TCF activated in set S.

Proposition 2. A memory test must force at least $v \cdot 2^v$ cell transitions in a set of v cells to be able to activate any fault that may affect the set of cells.

Proof: Consider the state transition diagram describing the triggering transitions in a set of v cells without faults. The memory test must force all the

transitions in this graph. Because the number of nodes in this graph of states is 2^v and from each node v arcs go to other adjacent nodes, the memory test must force $v \cdot 2^v$ different transitions to cover the graph of states.

Remark 2. Generally, the complex non-interacting faults are easier to detect than the primitive faults because there are more situations in which a complex fault is activated. Thus, if a memory test detects all primitive faults, it also detects all the complex non-interacting faults (the set of complex non-interacting faults dominates the set of primitive faults). However, for the interacting faults the combined effects of two primitive faults may cancel each other out before the affected cell is read again. More about interacting faults can be found in [1].

The march tests are the most popular and widely accepted deterministic tests because of their low temporal complexity, regular structures and their ability to detect a wide variety of memory faults.

Definition 4. A march element (M) consists of a sequence of operations applied to each cell in the memory before proceeding to the next cell. The whole memory is checked homogeneously in either one of two orders: ascending address order ($\Uparrow$) or descending order ($\Downarrow$). A march test is composed of m march elements, $<M^{(0)}; M^{(1)}; ...; M^{(m-1)}>$.

4 March Test for Reduced 3-Coupling

In this section a new march test *MT-R3CF* for the reduced model of 3-coupling is presented.

$$MT\text{-}R3CF = <\Updownarrow(w_0)^{(0)}; \Uparrow(rw_1)^{(1)}; \Uparrow(rw_0)^{(2)}; \Downarrow(rw_1)^{(3)}; \Downarrow(rw_0)^{(4)}; I_1{}^{(5)}; \Uparrow(rw_c rw_c)^{(6)}; I_2{}^{(7)}; \Uparrow(rw_c rw_c)^{(8)}; I_3{}^{(9)}; \Uparrow(rw_c rw_c)^{(10)}; I_4{}^{(11)}; \Uparrow(rw_c rw_c)^{(12)}; \Uparrow(r)^{(13)} >.$$

where I_1 , I_2 , I_3 , and I_4 are sequences which initialise the memory as follows: I_1 initialises the odd columns with 0 and the even columns with 1, and I_3 vice versa (column-stripe data background); I_2 and I_4 initialise the memory with a checkerboard data background and its complement (Fig. 2).

This march test contains fourteen sequences as identified with a superscript (x) where $x \in \{0,\ldots,13\}$. The test sequences (5)–(12) form an alternating series of background changes and march elements (as Cockburn proposed in [4]). Note that when changing from one background to the next, only the cells that must change states are written. Also, each write operation is preceded by a read operation. We can observe in Fig. 2 that any background change affects only a half of the cells. Each test sequence I_1, I_2, I_3, or I_4 performs $\frac{n}{2}$ read operations and $\frac{n}{2}$ write operations. Thus, *MT-R3CF* has a length of $30n$.

Theorem. The march test *MT-R3CF* detects all primitive reduced 3-coupling faults.

Proof: Consider an arbitrary triple set of cells $S=\{i,j,k\}$ that corresponds with one of the patterns P_1, P_2, P_3, P_4, P_5 and P_6. As shown in Fig. 1, we refer to the cells in S by i, j and k taking into account the order in which these cells are checked during the memory testing. Cells i, j and k are checked in this order when the memory is tested in ascending order ($\Uparrow$), and in reverse order, when the memory is tested in descending order ($\Downarrow$). As follows, we show that *MT-R3CF* activates and observes any primitive fault that affects the set of cells S.

a) *MT-R3CF activates any fault which affects a triple set of cells S.*

According to *Condition 1* previously defined, we must prove that *MT-R3CF* covers the Eulerian graph of states for a set of cells S, in all the six cases.

First, note that the sequence $\Updownarrow(w_0)$ loads into any triple set of cells the initial state <0,0,0>. Applying the march elements (1)–(4), *MT-R3CF* forces the transitions marked with solid lines in the Eulerian graph presented in Fig. 3, in every set of cells in the memory under test regardless of the pattern. The test sequences I_1, I_2, I_3 and I_4 ensure in a set of cells S, depending on the pattern (see Figs. 1 and 2), the initial states presented in Table 2. As highlighted in Table 2, the test sequences ensure the initial states <0,1,0> and <1,0,1> for all the six patterns.

In the Eulerian graph two adjacent nodes (states) have only one bit changed and two non-adjacent nodes have at least two bits changed. Applying the march $\Uparrow(rw_c rw_c)$ three adjacent nodes are visited and, finally, the set of cells returns to the initial state (the state that the sequence started with). The transitions forced by $\Uparrow(rw_c rw_c)$ in a set of cells S initialised with <0,1,0> and <1,0,1>, respectively, are marked with solid line in Fig. 4. The transitions highlighted in Figs. 3 and 4 show that the Eulerian graph of states is covered in all the six cases.

b) *MT-R3CF observes any primitive fault activated in set S.*

Table 3 presents the operations carried out in a set of cells $S=\{i,j,k\}$ during the memory testing, except on the writes for the first initialisation. The operations enclosed inside brackets sometimes are made, or sometimes are not made, depending on the set of cells. We can easily check in Table 3 that *Conditions 2* and *3* are also satisfied.

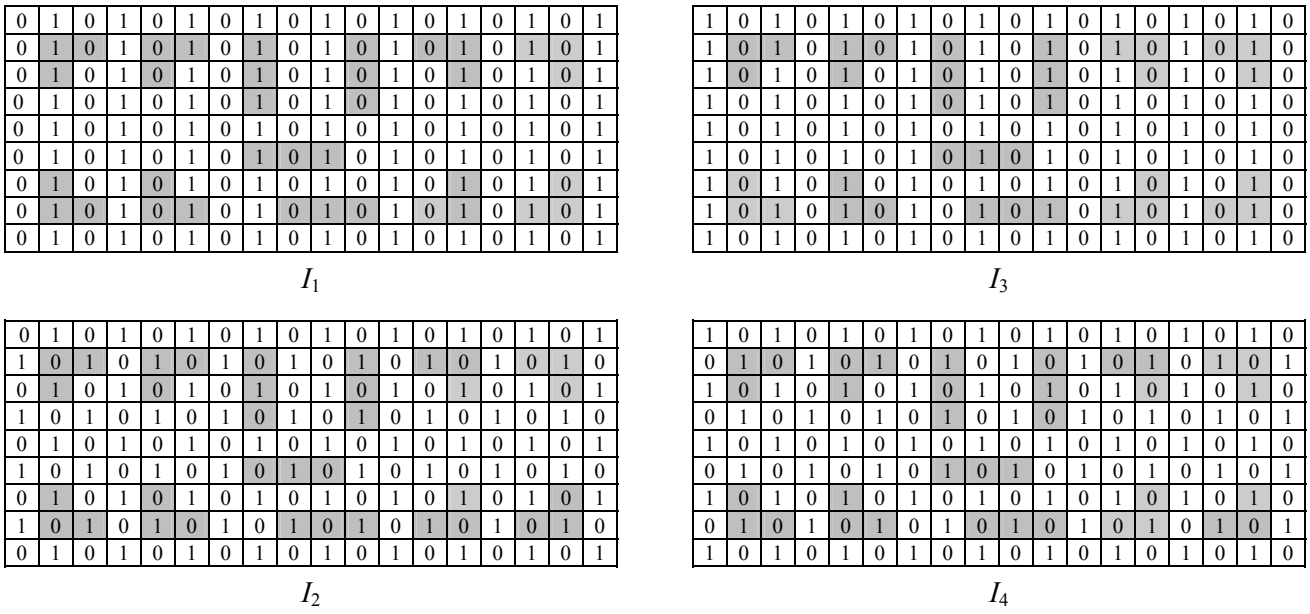

Fig. 2 – Data background used by *MT-R3CF* and the possible initial states for all six distinct patterns.

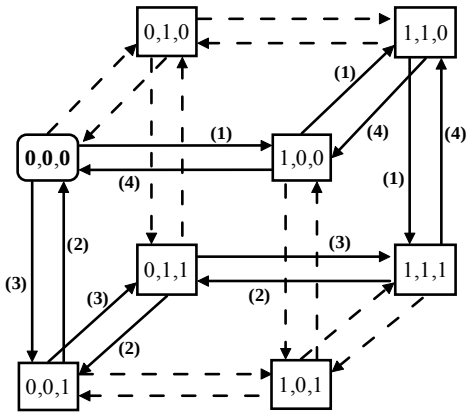

Fig. 3 – The Eulerian graph of states for a set of cells $S = \{i,j,k\}$ and the transitions carried out by the march elements (1) – (4).

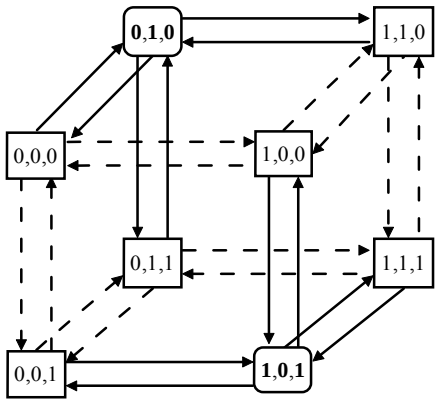

Fig. 4 – The transitions carried out in a set of cells $S = \{i,j,k\}$ by $\Uparrow(rw_c rw_c)$ for the initial states <0,1,0> and <1,0,1>.

Table 2–The initial states for a set of cells $S=\{i,j,k\}$.

	I_1 and I_3	I_2 and I_4
P_1	<**1**,**0**,**1**>, <**0**,**1**,**0**>	<0,1,1>, <1,0,0>
P_2	<1,1,0>, <0,0,1>	<**1**,**0**,**1**>, <**0**,**1**,**0**>
P_3	<**1**,**0**,**1**>, <**0**,**1**,**0**>	<1,1,0>, <0,0,1>
P_4	<0,1,1>, <1,0,0>	<**1**,**0**,**1**>, <**0**,**1**,**0**>
P_5	<**1**,**0**,**1**>, <**0**,**1**,**0**>	<0,1,0>, <1,0,1>
P_6	<0,0,0>, <1,1,1>	<**1**,**0**,**1**>, <**0**,**1**,**0**>

5 Simulation Results

To compare this test with other published march tests, simulation results regarding the ability of the tests to detect primitive 3-coupling faults are presented in this section. The following published tests have been considered for the simulation study:

- March test with 38*n* operations given by Caşcaval and Bennett [1] (*MT38n* in this paper);
- March test with 34*n*operations given by Caşcaval and Onea [2] (*MT34n* in this paper);
- Algorithm A with 30*n* operations given by Nair, Thatte, and Abraham [3] (*NTA*(*A*) in this paper);
- March test with 36*n* operations given by Papachristou and Sahgal [4] (*PS*(*A*) in this paper);
- *March C*– algorithm of length 10*n* and symmetric *March G* algorithm of length 24*n*, presented in [6].
- *March LR* of length 18*n* given by Yarmolik, van de Goor, Gaydadjiev and Mikitjuk [8];

The ability of these tests to detect the primitive reduced 3–coupling faults have been evaluated by simulation. Six sets of 3-coupled cells have been

Table 3 – The operations carried out in a set of cells $S=\{i,j,k\}$ by the march test *MT-R3CF*.

Operations	Comments
$\ldots\ r^i\ w_1^i \ldots\ r^j\ w_1^j \ldots\ r^k\ w_1^k \ldots\ r^i\ w_0^i \ldots\ r^j\ w_0^j \ \ldots\ r^k\ w_0^k\ \ldots$	March elements (1) and (2)
$\ldots\ r^k\ w_1^k \ldots\ r^j\ w_1^j \ldots\ r^i\ w_1^i \ldots\ r^k\ w_0^k \ldots\ r^j\ w_0^j \ \ldots\ r^i\ w_0^i\ \ldots$	March elements (3) and (4)
$\ldots [r^i\ w_c^i] \ldots [r^j\ w_c^j] \ldots [r^k\ w_c^k] \ldots$	Change to 2nd background (I_1)
$\ldots\ r^i\ w_c^i\ r^i\ w_c^i \ \ldots\ r^j\ w_c^j\ r^j\ w_c^j \ \ldots\ r^k\ w_c^k\ r^k\ w_c^k\ \ldots$	March element (6)
$\ldots [r^i\ w_c^i] \ldots [r^j\ w_c^j] \ldots [r^k\ w_c^k] \ldots$	Change to 3rd background (I_2)
$\ldots\ r^i\ w_c^i\ r^i\ w_c^i \ \ldots\ r^j\ w_c^j\ r^j\ w_c^j \ \ldots\ r^k\ w_c^k\ r^k\ w_c^k\ \ldots$	March element (8)
$\ldots [r^i\ w_c^i] \ldots [r^j\ w_c^j] \ldots [r^k\ w_c^k] \ldots$	Change to 4th background (I_3)
$\ldots\ r^i\ w_c^i\ r^i\ w_c^i \ \ldots\ r^j\ w_c^j\ r^j\ w_c^j \ \ldots\ r^k\ w_c^k\ r^k\ w_c^k\ \ldots$	March element (10)
$\ldots [r^i\ w_c^i] \ldots [r^j\ w_c^j] \ldots [r^k\ w_c^k] \ldots$	Change to 5th background (I_4)
$\ldots\ r^i\ w_c^i\ r^i\ w_c^i \ \ldots\ r^j\ w_c^j\ r^j\ w_c^j \ \ldots\ r^k\ w_c^k\ r^k\ w_c^k\ \ldots$	March element (12)
$\ldots\ r^i \ldots\ r^j \ \ldots\ r^k\ \ldots$	Final read sequence

Table 4. Fault coverage of simple reduced 3-coupling faults (expressed as %)

Test algorithm	*March C-*	*March LR*	*March G*	*NTA*(*A*)	*PS*(*A*)	*MT38n*	*MT34n*	*MT-R3CF*
Length	10*n*	18*n*	24*n*	30*n*	37*n*	38*n*	34*n*	30*n*
Fault coverage	50	62.5	62.5	63.89	63.89	94.91	96.33	100

considered, one for each pattern $P_i, i \in \{1, 2,3,4,5, 6\}$. For each set, all possible primitive 3-coupling faults have been simulated: 24 SCFs and 48 TCFs (see Table 1). In total, we have simulated 432 primitive faults. Simulation results regarding the ability of the tests to detect these primitive 3-coupling faults are presented in Table 4.

6 Final Remarks

The simulation results demonstrates the effectiveness of this march test when compared with other published tests. Remember that only the march test *MT-R3CF* detects all primitive reduced 3-coupling faults. To cover the model of reduced 3-coupling, *MT-R3CF* uses different data−background: a solid, a column−stripe, and a checkerboard.

The ability of this march test to detect dynamic faults will be the subject for upcoming paper.

References:

[1] Caşcaval, P., Bennett, S., Efficient March Test for 3-Coupling Faults in Random Access Memories, *Microprocessors and Microsystems*, Vol.24, No.10, 2001, pp.501-509.

[2] Caşcaval, P., Onea, A., March Test Algorithm for 3-Coupling Faults in Random Access Memories, WSEAS Press, Athens, 2002, pp. 188-194.

[3] Nair, R., Thatte, S., Abraham, J., Efficient Algorithms for Testing Semiconductor Random Access Memories, *IEEE Trans. Comput.*, Vol. C-27, No.6, 1978, pp.572-576.

[4] Papachristou, C., Sahgal, N., An Improved Method for Detecting Functional Faults in Semiconductor Random Access Memories, *IEEE Trans. Comput.*, C-34, 2, 1985, 110-116.

[5] Cockburn, B. F., Deterministic Tests for Detecting Single V-Coupling Faults in RAMs, *Journal of Electronic Testing – Theory and Applications*, Vol.5, No.1, 1994, pp.91-113.

[6] van de Goor, A.J., Using March Tests to Test SRAMs, *IEEE Design and Test of Computers*, March, 1993, pp.8-14.

[7] Suk, D., Reddy, S., Test Procedures for a Class of Pattern-Sensitive Faults in Semiconductor Random-Access Memories, *IEEE Trans. Comput.*, Vol. C-29, No.6, 1980, pp.419-429.

[8] Yarmolik, V.N., van de Goor, A.J., Gaydadjiev G.N., Mikitjuk, V.G., March LR: A test for realistic linked faults, *Proc. VLSI Test Symp.*, March, 1996, pp.272-280.