Testing of Inter-Word Coupling Faults in Word-Oriented SRAMs

X. Wang[1], M. Ottavi[2] and F. Lombardi[2]

[1] IBM Corp, Essex Junction (VT) USA.
[2] ECE Department, Northeastern University Boston (MA) USA.
E-mails: {lombardi, mottavi, xiawang}@ece.neu.edu

Abstract

A new algorithm to detect inter-word coupling faults in word-organized SRAMs (WOMs) is proposed in this paper. This algorithm (referred to as March-NU) relies on a new fault model which extends fault detection to three additional types of coupling faults, i.e. read destructive, deceptive read destructive and incorrect read coupling faults. These faults are related to well known fault mechanisms that have been reported in the literature to occur in the read operation of SRAMs. Previous algorithms can not guarantee 100% fault detection of these coupling faults. March-NU sensitizes and detects with 100 % coverage all coupling faults as well as traditional faults. A detailed analysis of its fault detection capabilities are presented. March-NU utilizes 8 March elements and its complexity is 30N, where N is the number of words in the WOM under test.

Index words: Coupling fault, memory, inter-word fault, testing, detection.

1: Introduction

Manufacturing of Static Random Access Memories (SRAMs) is a challenging process; in SRAMs [1], numerous defects (also known as physical faults) are possible. Through the years, many papers have established the relationships between physical and functional faults in these memories [2], [3] [1], [4], [5]. Different fault models have been proposed. In [1], the memory under test is divided into three parts to simplify fault modeling: faults can occur in memory cells, address decoders and read/write logic. Different faults have been identified as part of this model; the commonly analyzed faults are stuck-at (SAF), transition (TF), stuck-open (SOF), deceptive-read (DRF), coupling (CF), address decoder (AF), neighborhood pattern sensitive (NPSF). For fault modeling, memories can be differentiated by the addressing method: bit oriented (BOM) and word oriented memories (WOM) are possible. For BOM (WOM) read and write operations are independently (simultaneously) executed on each cell. The organization of the memory affects the execution of the test process. Memory testing consists of two basic procedures: to sensitize faults and detecting them. Faults can be sensitized by either forcing the cells into specified states, or performing some specific operations on the cells.

This paper deals with the detection of coupling faults; a coupling fault is the fault in which the operation (state) on the so-called coupling cells affects the operation of another (coupled) cell. In this paper, like in [1], coupling faults are restricted to two cells (i.e. one coupling cell and one coupled cell) therefore are a special case of k-coupling fault in which the coupling cells are k-1. The coupling cell is referred to as the *aggressor cell* (*a*-cell), while the coupled cell is referred to as the *victim cell* (*v*-cell). For a WOM, if the aggressor and the victim cells in a coupling fault are in different words (same word), then the coupling fault is referred to as *inter-word (intra-word)*.

Unlike single cell faults, coupling faults are related to the states or the operations of both the aggressor and the victim cells. In this case it has been shown that the so-called *fault mechanism*

Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'04)
1550-5774/04 $ 20.00 IEEE

(FME) describes the interactions between the coupling cell and the coupled cell and determines the sensitizing condition of the coupling faults. Detection of coupling faults can be accomplished by reading the states of the victim cells. For a coupling fault in a BOM and an inter-word coupling fault in a WOM, three types of FME have been proposed to describe the interaction between the coupling and coupled cells, because the aggressor (victim) cell is already in a state when the victim (aggressor) cell is affected by a read or write operation. The coupling faults which have been valided by simulation [6] [4] [7] are: State coupling faults (CFst), Disturb coupling faults (CFds), Transition coupling faults (CFtr), Write Destructive coupling faults (CFwd), Read Destructive coupling faults (CFrd), Incorrect Read coupling faults (CFir).

Few algorithms have been proposed for the detection of coupling faults in a WOM [4] [8] [9] [10]. The algorithm proposed in [4] can only detect the CFst coupling fault. The algorithm in [10] can only detect intra-word coupling faults of CFst and CFds. [9] can detect intra-word coupling faults of idempotent coupling faults (CFid), CFds, and CFst. [9] has shown that CFid is a subset of CFds. [8] can detect intra-word coupling faults of CFst, CFds, CFtr, CFwd, CFrd, and CFir. [4], [10], [9] have short test length, but the fault coverage is not as complete as the algorithm of [8].

The objective of this paper is to provide a new algorithm to test inter-word coupling faults in two cells of a WOM. This algorithm (referred to as March-NU) relies on a new fault model which extends fault detection to three new types of coupling faults, i.e. read destructive, deceptive read destructive and incorrect read coupling faults. These faults are related to well known fault mechanisms that have been proved by simulation to occur in the read operation. Moreover differently from previous works, the analysis shows that March-NU is applicable to both BOM and WOM. March-NU utilizes 8 March elements and its complexity is $30N$, where N is the number of words in the WOM under test. March-NU can sensitize and detect all coupling faults as well as traditional faults (such as stuck-at and transition faults, for example).

This paper is organized as follows. Section 2 deals with preliminaries inclusive of notation. Section 3 presents the inter-word coupling fault model; a new extended fault model is also outlined. The proposed approach to testing of inter-word coupling faults is detailed in Section 4; basic properties, March elements and fault detection are discussed in detail. Section 5 compares the proposed approach to existing algorithms. Conclusions are given in Section 6.

2: Preliminaries

The notation used in this paper follows previous works.

State: S denotes the state of a cell. $S \in \{0, 1, \uparrow, \downarrow, d\}$; "$\uparrow$" denotes the transition state from "0" to "1"; "$\downarrow$" denotes the transition state from "1" to "0", "d" denotes the "do not care" state.

Operation: O denotes the operation on a cell. $O \in \{0w0, 0w1, 1w0, 1w1, r0, r1\}$ where w (r) stands for the write (read) operation. For example, "$0w0$" denotes an operation that writes "0" to a cell (final state) which is in an original state given by "0".

Sensitizing Condition: The sensitizing condition is the state (or operation) of the aggressor and victim cells that sensitizes coupling faults. C denotes the sensitizing condition. $C \in \{0, 1, 0w0, 0w1, 1w0, 1w1, r0, r1\}$. C_a denotes the sensitizing condition of the aggressor cell; C_v denotes the sensitizing condition of the victim cell.

Sensitizing Condition Pair: the sensitizing condition of both the a-cell (aggressor) and v-cell (victim). (C_a, C_v) denotes a sensitizing condition pair. For example, $(0, 1)$ means that when the a-cell is "0" and the v-cell is "1", there will be a coupling fault; $(0w0, 1)$ means that when the a-cell is executing the operation "$0w0$" and the v-cell is "1", then there could be a coupling fault.

Addresses: "$a > v$" denotes that in memory the *address* of the a-cell is *higher* than the address of the v-cell; "$a < v$" means the address of the a-cell is it lower than the address of the v-cell.

As in previous works, unlinked faults are assumed, i.e. a cell is affected by only one type of fault.

Fault Mechanism(FME): FME describes the interaction between the coupling and victim cells. There are four types of FMEs and a FME can be expressed by Fault Primitives (FP). A type of FME consists of a set of FPs.

Fault Primitive (FP): the definition of FP is as in [8], [9], [10], i.e. a FP describes the sensitizing condition of the aggressor and victim cells. A coupling fault consists of a set of FPs; $< (C_a, C_v); S_c; S_r; S_o >$ denotes a FP in which (C_a, C_v) denotes the sensitizing condition pair of the aggressor and victim cells. S_c denotes the correct state that the victim cell is supposed to be in the absence of a fault. S_r denotes the actual state of the victim cell. S_o denotes the state of the output. If the output is "do not care", then S_o will have "d" as value. The state of the output and the state of the victim cell must be specified because the state of the output will be different from the state of the victim cell for some faults (such as the Deceptive Read Destructive coupling fault (CFdr) and Incorrect read coupling fault (CFir) [10]).

Fault Model(FMO): FMO describes the interaction between the coupling and coupled cells. A type of FME may consists of more than one types of FMO (a type of FMO must belong to only one type of FME). There is a similarity between FME and FMO, i.e. both FME and FMO can be expressed by the a set of FPs. The only difference between FME and FMO is that a type of FMO may consist of a set of FPs (which is a subset of a type of FME).

3: Inter-word Coupling Fault Model

For a WOM, if the aggressor and the victim cells of a coupling fault are in different words, then this fault is referred to as a inter-word coupling fault.

3.1: Fault Mechanisms

In [7] the characterization of three FMEs for inter-word coupling faults has been provided :

1) A_sV_s: The state of the aggressor cell sensitizes the fault in the victim cell and makes the victim cell to be in a faulty state.

2) A_oV_s**:** The operation of the aggressor cell sensitizes the fault in the victim cell and makes the victim cell to be in a faulty state.

3) A_sV_o**:** The state of the aggressor cell and the operation of the victim cell sensitize the fault in the victim cell and makes the victim cell to be in a faulty state.

Table 1 reports all possible FPs for each of these FMEs.

3.2: Extended Coupling Fault Model

Table 1 reports the extended fault model considered. The basic set of coupling faults [4] [7] is as follows:

State Coupling Fault (CFst): If the a-cell is in a state, then the v-cell is forced into a given state, by performing no operation (on the v-cell or a-cell). CFst belongs to A_sV_s. CFst consists of 4 FPs.

Disturb Coupling Fault (CFds): An operation (write or read) performed on the a-cell causes the v-cell to be in a faulty state. CFds consists of the 12 FPs of $AoVs$

Transition Coupling Faults (CFtr): A given logic value in the a-cell results in a write operation with a failing transition for the v-cell. This fault is sensitized by first placing the a-cell to a given state, and thereafter by applying a write operation to the v-cell. CFtr belongs to the A_sV_o FME and consists of its first 4 FPs (FP1 through FP4).

Write Destructive coupling fault (CFwd): A given logic value in the a-cell results in a failing write operation for the v-cell. This fault is sensitized by first placing the a-cell to a given state, and thereafter by applying a write operation to the v-cell. CFwd belongs to the A_sV_o FME and consists of 4 FPs (FP5 through FP8)

The previous fault model set is extended by considering the fault detection of the following coupling faults as reported by Hamdioui [6].

Fault Mechanism	Number	Description	Fault
A_sV_s	1	$<(0,0);0;1;d>$	CFst
	2	$<(0,1);1;0;d>$	
	3	$<(1,0);0;1;d>$	
	4	$<(1,1);1;0;d>$	
A_oV_s	1	$<(0w0,0);0;1;d>$	CFds
	2	$<(0w0,1);1;0;d>$	
	3	$<(0w1,0);0;1;d>$	
	4	$<(0w1,1);1;0;d>$	
	5	$<(1w0,0);0;1;d>$	
	6	$<(1w0,1);1;0;d>$	
	7	$<(1w1,0);0;1;d>$	
	8	$<(1w1,1);1;0;d>$	
	9	$<(r0,0);0;1;d>$	
	10	$<(r0,1);1;0;d>$	
	11	$<(r1,0);0;1;d>$	
	12	$<(r1,1);1;0;d>$	
A_sV_o	1	$<(0,0w1);1;0;d>$	CFtr
	2	$<(0,1w0);0;1;d>$	
	3	$<(1,0w1);1;0;d>$	
	4	$<(1,1w0);0;1;d>$	
	5	$<(0,0w0);0;1;d>$	CFwd
	6	$<(0,1w1);1;0;d>$	
	7	$<(1,0w0);0;1;d>$	
	8	$<(1,1w1);1;0;d>$	
	9	$<(0,r0);0;1;1>$	CFrd
	10	$<(0,r1);1;0;0>$	
	11	$<(1,r0);0;1;1>$	
	12	$<(1,r1);1;0;0>$	
	13	$<(0,r0);0;1;0>$	CFdr
	14	$<(0,r1);1;0;1>$	
	15	$<(1,r0);0;1;0>$	
	16	$<(1,r1);1;0;1>$	
	17	$<(0,r0);0;0;1>$	CFir
	18	$<(0,r1);1;1;0>$	
	19	$<(1,r0);0;0;1>$	
	20	$<(1,r1);1;1;0>$	

Table 1. All possible FPs of A_sV_s A_oV_s A_sV_o

Read Destructive Coupling Fault (CFrd): A given logic value in the a-cell results in a destructive read operation for the v-cell; this forces the v-cell into a faulty state and produces an erroneous output. This fault is sensitized by first placing the a-cell in a given state, and thereafter by applying a read operation to the v-cell. CFrd belongs to the A_sV_o FME and consists of the 4 FPs FP9 through FP12.

Deceptive Read Destructive Coupling Fault (CFdr): A given logic value in the a-cell results in a destructive read operation to the v-cell; this forces the v-cell into a faulty state, but it will produce a correct output. This fault is sensitized by first setting the a-cell in a state, and thereafter by applying a read operation to the v-cell. CFdr belongs to the A_sV_o FME and consists of its 4 FPs FP13 through FP16.

Incorrect Read Coupling Fault (CFir): A logic value in the a-cell results in an incorrect read operation for the v-cell; this does not affect the state of the v-cell but it will produce the wrong output. This fault is sensitized by first setting the a-cell in a state, and thereafter by applying a read operation to the v-cell. CFir belongs to A_sV_o and consists of its 4 FPs FP17 through FP20.

4: Proposed Test Approach

4.1: Basic Properties

The fault mechanisms A_sV_s, A_oV_s and A_sV_o can describe the interaction between the coupling and coupled cells for inter-word coupling faults in a WOM. Moreover a WOM test algorithm reads and writes a word (which is referred to as the data pattern); in particular, a data backgroud (DB) is a one word test pattern [4], the width of a DB is given by B and defines the width of the word of the WOM. The detection of coupling faults in a WOM consists of a sequence of read or write operations with DB. Define D_0 as the DB made of all "0", D_1 as the DB made of all "1" (the width of D_0 or D_1 is B).

To derive the detection algorithm for all inter-word coupling faults (inclusive of the three additional types of coupling faults in the extended fault model), the test sequence of each type of faults is initially established; all test sequences are then merged and redundant sequences are deleted.

The proposed testing algorithm is referred to as March-NU (NU:Northeastern University).

March-NU can detect all inter-word coupling faults (i.e. CFst, CFds, CFtr, CFwd, CFrd, CFdr, CFir). March-NU can also detect SAF, TF, and AF. The March-NU algorithm is an improvement over previous works such as the BOM based March algorithms of [11], [1] as these algorithms can not detect all inter-word coupling faults such as CFrd, CFdr, CFir.

4.2: March Elements

Table 2 describes the *March elements* of March-NU. There are 8 March elements, and the test length of the proposed algorithm (denoted by L) is $30N$, where N is the number of words in the WOM-based memory under test. The function of each March element of March-NU is as follows:

M0: The cells are initialized to "0". The operation sequence of M0 is $\Updownarrow(wD_0)$. The cells can be accessed either from the lowest to the highest address, or reverse (*increasing or decreasing order*).

M1: The operation sequence of M1 is:
$\Uparrow(rD_0, wD_0, rD_0, rD_0, wD_1, rD_1, wD_1, rD_1, rD_1, wD_0, rD_0)$.
The cells must be accessed in increasing address order. There are 11 operations which are denoted as op$i$, i=0, ... , 10. During M1, each cell in the corresponding words will experience the sequence "$r0, 0w0, r0, r0, 0w1, r1, 1w1, r1, r1, 1w0, r0$." For inter-word coupling faults, the operation sequence will sensitize and detect some of the FPs of CFst, CFds, CFtr, CFwd, CFrd, CFdr, CFir. For single cell faults, the operation sequence sensitizes and detects stuck at (SAF) and transition faults (TF).

M2: The operation sequence of M2 is $\Uparrow(rD_0)$. The operation will detect some of the FPs of CFst, CFds, CFtr, CFwd, CFrd, CFdr, CFir which were sensitized by M1.

M3 and M4: These elements are used to test address faults (AF).

M5: This element is similar to M0. The cells are initialized to "1". The operation sequence of M5 is $\Updownarrow(wD_1)$. The cells can be accessed in either increasing or decreasing address order.

M6: This element is similar to M1. The operation sequence of M6 is:
$\Downarrow(rD_1, wD_0, rD_0, rD_0, wD_0, rD_0, wD_1, rD_1, rD_1, wD_1, rD_1)$.
The cells must be accessed in decreasing address order. There are 11 operations; in M6, each cell of the corresponding words will experience the operation sequence

"$r1, 1w0, r0, r0, 0w0, r0, 0w1, r1, r1, 1w1, r1$" For inter-word coupling faults, the operation sequence will sensitize and detect some of the FPs of CFst, CFds, CFtr, CFwd, CFrd, CFdr, CFir. For single cell faults, the operation sequence can sensitize and detect stuck at (SAF) and transition faults(TF).

M7: This element is similar to M2. The operation sequence of M7 is $\Downarrow(rD_1)$. The operation detects some of the FPs of CFst, CFds, CFtr, CFwd, CFrd, CFdr, CFir which were sensitized by M6.

March Element	Operation Sequence	Test Length
M0	$\Updownarrow(wD_0)(init)$	$1N$
M1	$\Uparrow(rD_0, wD_0, rD_0, rD_0, wD_1, rD_1, wD_1, rD_1, rD_1, wD_0, rD_0)$	$11N$
M2	$\Uparrow(rD_0)$	$1N$
M3	$\Uparrow(rD_0, wD_1)$	$2N$
M4	$\Downarrow(rD_1, wD_0)$	$2N$
M5	$\Updownarrow(wD_1)$	$1N$
M6	$\Downarrow(rD_1, wD_0, rD_0, rD_0, wD_0, rD_0, wD_1, rD_1, rD_1, wD_1, rD_1)$	$11N$
M7	$\Downarrow(rD_1)$	$1N$
L		$30N$

Table 2. March-NU (N is the number of words)

4.3: Fault Detection of Coupling Faults

The inter-word coupling faults which have been analyzed in previous papers are also detected in March-NU; fault sensitizing and detection of these faults is obtained as described in Table 3 for CFst, CFds, CFtr and CFwd.

Fault Type	No.	FP	Address	Sensitized by	Detected by
CFst	1	$<(0,0);0;1;d>$	$a><v$	op0 of M0	op0 of M1
	2	$<(0,1);1;0;d>$	$a><v$	op4 of M1	op5 of M1
	3	$<(1,0);0;1;d>$	$a><v$	op1 of M6	op2 of M6
	4	$<(1,1);1;0;d>$	$a><v$	op6 of M6	op7 of M6
CFds	1	$<(0w0,0);0;1;d>$	$a>v$	op1 of M1	op0 of M2
	1	$<(0w0,0);0;1;d>$	$a<v$	op1 of M1	op2 of M1
	2	$<(0w0,1);1;0;d>$	$a>v$	op4 of M6	op5 of M6
	2	$<(0w0,1);1;0;d>$	$a<v$	op4 of M6	op0 of M7
	3	$<(0w1,0);0;1;d>$	$a>v$	op4 of M1	op0 of M2
	3	$<(0w1,0);0;1;d>$	$a<v$	op4 of M1	op5 of M1
	4	$<(0w1,1);1;0;d>$	$a>v$	op6 of M6	op7 of M6
	4	$<(0w1,1);1;0;d>$	$a<v$	op6 of M6	op0 of M7
	5	$<(1w0,0);0;1;d>$	$a>v$	op9 of M1	op0 of M2
	5	$<(1w0,0);0;1;d>$	$a<v$	op9 of M1	op10 of M1
	6	$<(1w0,1);1;0;d>$	$a>v$	op1 of M6	op2 of M6
	6	$<(1w0,1);1;0;d>$	$a<v$	op1 of M6	op0 of M7
	7	$<(1w1,0);0;1;d>$	$a>v$	op6 of M1	op0 of M2
	7	$<(1w1,0);0;1;d>$	$a<v$	op6 of M1	op7 of M1
	8	$<(1w1,1);1;0;d>$	$a>v$	op6 of M6	op7 of M6
	8	$<(1w1,1);1;0;d>$	$a<v$	op6 of M6	op0 of M7
	9	$<(r0,0);0;1;d>$	$a>v$	op10 of M1	op0 of M2
	9	$<(r0,0);0;1;d>$	$a<v$	op10 of M1	op0 of M1
	10	$<(r0,1);1;0;d>$	$a>v$	op5 of M6	op0 of M6
	10	$<(r0,1);1;0;d>$	$a<v$	op5 of M6	op0 of M7
	11	$<(r1,0);0;1;d>$	$a>v$	op10 of M1	op0 of M2
	11	$<(r1,0);0;1;d>$	$a<v$	op10 of M1	op0 of M1
	12	$<(r1,1);1;0;d>$	$a>v$	op10 of M6	op0 of M6
	12	$<(r1,1);1;0;d>$	$a<v$	op10 of M6	op0 of M7
CFtr	1	$<(0,0w1);1;0;d>$	$a><v$	op4 of M1	op5 of M1
	2	$<(0,1w0);0;1;d>$	$a><v$	op9 of M1	op10 of M1
	3	$<(1,0w1);1;0;d>$	$a><v$	op6 of M6	op7 of M6
	4	$<(1,1w0);0;1;d>$	$a><v$	op1 of M6	op2 of M6
CFwd	1	$<(0,0w0);0;1;d>$	$a><v$	op1 of M1	op2 of M1
	2	$<(0,1w1);1;0;d>$	$a><v$	op6 of M1	op7 of M1
	3	$<(1,0w0);0;1;d>$	$a><v$	op4 of M6	op5 of M6
	4	$<(1,1w1);1;0;d>$	$a><v$	op9 of M6	op10 of M6

Table 3. Test sequence for inter-word CFst, CFds, CFtr and CFwd of a WOM

The detailed treatment of these test sequences is omitted due to lack of space; moreover, these faults can be already detected by previous algorithms.

The sensitizing and detection processes for the extended coupling model (as analyzed in this paper) addresses three additional faults: CFrd, CFdr and CFir. Table 4 shows the sensitizing and detecting of each FP of these three additional coupling faults.

A detailed description of these test sequences is now provided.

Consider initially **CFrd**, the test process involved for each FP is as follows:

To test FP1 $<(0,r0);0;1;1>$ with $a>v$ and $a<v$, op2 of M1 is used; "rD_0" makes each cell of the corresponding word to read a "0" (the other words are already "0"). The state (or operation)

Fault Type	FP No.	FP	Address	Sensitized by	Detected by
CFrd	1	$<(0,r0);0;1;1>$	$a><v$	op2 of M1	op2 of M1
	2	$<(0,r1);1;0;0>$	$a><v$	op7 of M1	op7 of M1
	3	$<(1,r0);0;1;1>$	$a><v$	op2 of M6	op2 of M6
	4	$<(1,r1);1;0;0>$	$a><v$	op7 of M6	op7 of M6
CFdr	1	$<(0,r0);0;1;0>$	$a><v$	op2 of M1	op3 of M1
	2	$<(0,r1);1;0;1>$	$a><v$	op7 of M1	op8 of M1
	3	$<(1,r0);0;1;0>$	$a><v$	op2 of M6	op3 of M6
	4	$<(1,r1);1;0;1>$	$a><v$	op7 of M6	op8 of M6
CFir	1	$<(0,r0);0;0;1>$	$a><v$	op2 of M1	op2 of M1
	2	$<(0,r1);1;1;0>$	$a><v$	op7 of M1	op7 of M1
	3	$<(1,r0);0;0;1>$	$a><v$	op2 of M6	op2 of M6
	4	$<(1,r1);1;1;0>$	$a><v$	op7 of M6	op7 of M6

Table 4. Test sequence of inter-word CFrd CFdr CFir of a WOM

Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'04)
1550-5774/04 $ 20.00 IEEE

of the a-cell (and the v-cell) is $< 0, r0 >$, so FP1 $< (0, r0); 0; 1; 1 >$ is sensitized and the state of the victim cell is ”1”; op2 of M1 ”rD_0” can detect the faulty state of the victim cells.

To test FP2 $< (0, r1); 1; 0; 0 >$ with $a > v$ and $a < v$, op7 of M1 is used. For ”rD_1” the state (or operation) of the a-cell (and the v-cell) is $< 0, r1 >$, so FP2 $< (0, r1); 1; 0; 0 >$ is sensitized and the state of victim cell is ”1”; op7 of M1 ”rD_0” can detect the faulty state of the victim cells.

To test FP3 $< (1, r0); 0; 1; 1 >$ with $a > v$ and $a < v$, op2 of M6 is used. For ”rD_0” the state (or operation) of the a-cell (and the v-cell) is $< 1, r0 >$, so FP2 $< (1, r0); 0; 1; 1 >$ is sensitized and the state of victim cell is ”1”; op2 of M6 ”rD_0” can detect the faulty state of the victim cells.

To test FP4 $< (1, r1); 1; 0; 0 >$ with $a > v$ and $a < v$, op7 of M6 is used. For ”rD_1” the state (or operation) of the a-cell (and the v-cell) is $< 1, r1 >$, so FP4 $< (1, r1); 1; 0; 0 >$ is sensitized and the state of victim cell is ”1”; op7 of M6 ”rD_1” can detect the faulty state of the victim cells.

The process of sensitizing and detecting each FP of **CFdr** is as follows.

To test FP1 $< (0, r0); 0; 1; 0 >$ with $a > v$ and $a < v$, op2 of M1 ”rD_0” is used; this ensures that the states (or operations) of the a-cell and the v-cell are $< 0, r0 >$, so FP1 $< (0, r0); 0; 1; 0 >$ is sensitized. The state of the victim cell is the faulty state ”1” while the output is the correct state ”0”; op3 of M1 ”rD_0” can detect the faulty state of the victim cells.

To test FP2 $< (0, r1); 1; 0; 1 >$ with $a > v$ and $a < v$, op7 of M1 (i.e. ”rD_1” is used to make each cell of the corresponding word to have ”$r1$” as operation (the other words are already ”0”), the states (or operations) of the a-cell and v-cell are $< 0, r1 >$, so FP2 $< (0, r1); 1; 0; 1 >$ is sensitized. The state of the victim cell is the faulty state ”0” while the output is the correct state ”1”, op8 of M1 ”rD_1” can detect the faulty state of the victim cells.

To test FP3 $< (1, r0); 0; 1; 0 >$ with $a > v$ and $a < v$, in March-NU op2 of M6 given by ”rD_0” is used. The states (or operations) of the a-cell and v-cell are $< 1, r0 >$, so FP3 $< (1, r0); 0; 1; 0 >$ is sensitized. The state of the victim cell is the faulty state ”1” while the output is the correct state ”0”, op3 of M6 ”rD_0” can detect the faulty state of the victim cells.

To test FP4 $< (1, r1); 1; 0; 1 >$ with $a > v$ and $a < v$,March-NU uses op7 of M6 ”rD_1”. The states (or operations) of the a-cell and the v-cell are $< 1, r1 >$, so FP4 $< (1, r1); 1; 0; 1 >$ is sensitized. The state of the victim cell is the faulty state ”0” while the output is the correct state ”1”, op8 of M6 ”rD_1” can detect the faulty state of the victim cells.

Finally, each FP of **CFir** must be considered.

To test FP1 $< (0, r0); 0; 0; 1 >$ with $a > v$ and $a < v$, op2 of M1 (i.e. ”rD_0”) is used. The state (or operation) of the a-cell (and the v-cell) is $< 0, r0 >$, so FP1 $< (0, r0); 0; 0; 1 >$ is sensitized. The state of the victim cell is the correct state ”0” while the output is the faulty state ”1”, op2 of M1 ”rD_0” can detect the faulty state of the victim cells.

To test FP2 $< (0, r1); 1; 1; 0 >$ with $a > v$ and $a < v$, op7 of M1 (i.e. ”rD_1”) is utilized. The state (or operation) of the a-cell (and the v-cell) is $< 0, r1 >$, so FP2 $< (0, r1); 1; 1; 0 >$ is sensitized. The state of the victim cell is the correct state ”1” while the output is the faulty state ”0”, op7 of M1 ”rD_1” can detect the faulty state of the victim cells.

To test FP3 $< (1, r0); 0; 0; 1 >$ with $a > v$ and $a < v$, op2 of M6 (i.e. ”rD_0”) is used. The state (or operation) of the a-cell (and the v-cell) is $< 1, r0 >$, so FP3 $< (1, r0); 0; 0; 1 >$ is sensitized. The state of the victim cell is the correct state ”0” while the output is the faulty state ”1”, op2 of M6 ”rD_0” can detect the faulty state of the victim cells.

To test FP4 $< (1, r1); 1; 1; 0 >$ with $a > v$ and $a < v$, op7 of M6 (i.e. ”rD_1”) is utilized. The state (or operation) of the a-cell (and the v-cell) is $< 1, r1 >$, so FP4 $< (1, r1); 1; 1; 0 >$ is sensitized. The state of the victim cell is the correct state ”1” while the output is the faulty state ”0”, op7 of M6 ”rD_1” can detect the faulty state of the victim cells.

4.4: Fault Detection of Other Faults

As for other faults in the memory under test, detection using March-NU is accomplished as follows.

Condition	March Element
1	$\Uparrow (rx, ..., w\bar{x})$
2	$\Downarrow (r\bar{x}, ..., wx)$

Table 5. Conditions for detecting address decoder faults

- In M1 and M6 of March-NU, the operations of write "0", read "0", write "1", read "1" are employed, so SAF (both at "0" and "1") can be sensitized and detected.
- In M1 and M6, state transitions from "0" to "1" and from "1" to "0" occur; therefore, TF (trnsition fault) can be sensitized and detected by March-NU.
- In [1], the conditions to sensitize and detect the address decoder fault (AF) were provided. These conditions are also shown in Table 5. In Table 5, x is either "0" or "1" while $\bar{x}$ denotes the complement of x. For Condition 1, the operation sequence must start as rx and end as $w\bar{x}$; any other operation sequence can be executed between rx and $w\bar{x}$¿ For Condition 2, the operation sequence must start as $r\bar{x}$ and end as wx; any other operation sequence can be executed between $r\bar{x}$ and wx In March-NU, M3 and M4 satisfy these conditions, so AF can be sensitized and detected by March-NU.
- In [1], the conditions to sensitize and detect the address stuck-open fault (SOF) were provided as follows:
  "$rx, ..., w\bar{x}, r\bar{x}$"; M1 and M6 in March-NU satisfy these conditions, so SOF can be sensitized and detected.
- To detect a deceptive read fault (DRF) [5] two consecutive reading operations are required. The first read operation will generate the correct result; the second read operation will generate the faulty result. The consecutive read operations "rD_0, rD_0" and "rD_1, rD_1" in M1 and M6 of March-NU can sensitize and detect the deceptive read fault (DRF).

Hence, March-NU can detect SAF, TF, AF, DRF and inter-word coupling faults (of type CFst, CFds, CFtr, CFwd, CFrd, CFdr and CFir). The test length of March-NU is $30N$ where N is the number of words in the memory.

March-NU can also be applied to BOMs (i.e. D_0 is replaced by "0" and D_1 is replaced by "1".); in this case the complexity is $30B$ where B is the number of bits of the tested memory. The sensitizing and detection processes of faults in a BOM are the same as previously described.

5: Comparison

March-NU can be compared with previous test algorithms; this comparison can only be made with respect to BOM as none of the previous algorithms are applicable to WOM. This is shown in Table 6. In this table, M denotes the organization (i.e. B is for BOM and W is for WOM), "+" denotes that the algorithm has 100% coverage of the specified type of fault. "-" denotes that the algorithm does not have 100% coverage of the specified type of fault. While March-NU has coverage of all faults, previous algorithms [12] are only applicable to BOM and fail to achieve complete coverage. In Table 6, the test length of March-G is a function of D, where D denotes the number of data retention delay operations.

6: Conclusions

This paper has presented a new model applicable to coupling faults in Word-Oriented SRAMs. An algorithm with full coverage of the inter-word coupling faults and traditional faults (such as SAF, TF and AF for example) has been developed. This algorithm detects inter-word coupling faults in two cells of a WOM and is referred to as March-NU. It relies on a new fault model which extends

Approach	M	L	AF, SAF,TF	DRF	SOF	CFst, CFds, CFtr, CFwd	CFrd, CFdr,CFir
[1]	B	$10B$	+	-	+	+	-
[13]	B	$23B$+2D	+	+	+	+	-
[14]	B	$14B$	+	-	+	+	-
[15]	B	$13B$	+	-	+	+	-
[11]	B	$22B$	+	-	+	+	-
NU	B/W	$30B(N)$	+	+	+	+	+

Table 6. Comparision of fault coverage of unlinked faults of March-NU and previous test algorithms

fault detection to three new types of coupling faults, i.e. read destructive, deceptive read destructive and incorrect read coupling faults. These faults are related to well known fault mechanisms that have been proved by simulation to occur in the read operation. Moreover differently from previous works, the analysis shows that March-NU is applicable to both BOM and WOM. March-NU utilizes 8 March elements and its complexity is $30N$, where N is the number of words in the WOM under test.

References

[1] van de Goor. A.J, "Testing semiconductor memories, theory and practice," in *John Wiley and Sons*, Chichester, UK, 1991.

[2] A.N. Mclaren, C.H. Stapper, and M. Dreckmann, "Yield model for productivity," in *IBM J. Res. Develop.*, 1980, vol. 24, pp. 398 – 409.

[3] C.H. Stapper, "Modeling of integrated circuit defect sensitivities," in *IBM J. Res. Develop.*, 1983, vol. 27, pp. 549–557.

[4] R. Dekker, F. Beenker, and L. Thijssen, "A realistic fault model and test algorithms for static random access memories," in *IEEE Trans. on Computer-Aided Design*, 1990, vol. 9, pp. 567 – 572.

[5] R.D. Adams., "Extensions of static random access memories, fault modeling and examination of patterns for fault detection," in *Master Thesis, Dartmouth College*, 1996.

[6] Hamdioui S. and A. J. van de Goor, "Experimental analysis of spot defects in srams: Realistic fault models and tests," in *Proc. Ninth Asian Test Symposium ATS'2000*, 2000, pp. 131 – 138.

[7] S. Hamdioui, "Testing multi-port memories: Theory and practice," in *PHD. Thesis, Delft University*, 2001, vol. ISBN 90-9014986-4, p. 100.

[8] A.J. van de Goor, S. Hamdioui, and M. Rodgers, "Detecting intra-word faults in word-oriented memories," in *IEEE VLSI Test Symposium*, Napa Valley, CA, USA, 2003.

[9] A.J. van de Goor and I.B.S. Tlili, "March tests for word-oriented memories," in *Proc. of Design Automation and Test in Europe*, 1998, pp. 501 – 508.

[10] I.B.S Tlili, A.J van de Goor, and S. Hamdioui, "Converting march test for bit-oriented memories into tests for word-oriented memories," in *Proc. of IEEE Int. Workshop on Memory Technology, Design and Testing*, 1998, pp. 46 – 52.

[11] van de Goor. A.J et al, "March la: A test for linked memory faults," in *Proc. European Design and Test Conf.*, Paris, France, 1991, pp. 627–627.

[12] van de Goor. A.J, "Embedded(core) memory testing: Fault models, algorithms, tests and industrial results," in *Tutorial at IBM*, Burlington, VT, 8/2001.

[13] A.J. van de Goor, "Using march tests to test srams," in *IEEE Design and Test of Computers*, March 1993, vol. 10.

[14] V.N. Yarmolik, A.J van de Goor, G.N. Gaydajiev, and Mikitjuk V.G., "March lr: A test for realistic linked faults," in *Proc. of 14-th IEEE VLSI Test Symposium*, 1996, pp. 272 – 280.

[15] Gaydadjiev G.N. and A.J. van de Goor, "March u: A test for unlinked memory faults," in *IEEE Proceedings Circuits, Devices and Systems*, June 1997, vol. 144.