# Detecting Intra-Word Faults in Word-Oriented Memories

Said Hamdioui[1,2] Ad J. van de Goor[2] Mike Rodgers[1]
[1]Intel Corporation, 2200 Mission College Boulevard, Santa Clara, CA 95052
[2]Delft University of Technology, Faculty of Information Technology and Systems
Computer Engineering Laboratory, Mekelweg 4, 2628 CD Delft, The Netherlands
E-mail: said@ce.et.tudelft.nl

## Abstract

*This paper improves upon the state of the art in testing word oriented memories. It first presents a complete set of fault models for intra-word coupling faults. Then, it establishes the data background sequence (DBS) for each intra-word coupling fault. These DBSs will be compiled into a $(1+28*\lfloor log_2B\rfloor)*\frac{n}{B}$ test with complete fault coverage of the target faults, where $n$ is the size of the memory and $B$ the word size. The test length can be reduced to $29*\frac{n}{B}$ when the intra-word faults are restricted to physical adjacent cells within a word.*

**Keywords:** *Bit-oriented/ word-oriented memories, memory tests, data backgrounds, fault models.*

## 1 Introduction

Random access memories can be organized as *bit-oriented* memories (BOMs) or as *word-oriented* memories (WOMs). WOMs contain more then one bit per word; i.e., $B > 2$, whereby $B$ represents the number of bits per word, and usually is a power of 2.

Traditionally, WOMs have been tested by repeated application of BOM tests, where different *data backgrounds (DBs)* are used during each iteration. The disadvantages of this methodology are test time inefficiency and limited fault coverage of coupling faults between cells within the same word, which are called *intra-word faults*. In [3], a systematic method to solve the problem of detecting intra-word faults has been designed. The solution was based on observing that most march tests contain read and write operations with some data values as well as the complementary data values. In [10], a method based on $m$-out-of-$n$ codes has been presented to derive an optimal set of data backgrounds. However, the solutions reported in [3, 10] were restricted only to *state coupling faults*. Currently published work shows the existence of many new *coupling faults* [2, 4]. In [5] the transformation of BOM tests into WOM tests has been established, based on replacing the data values used by the march tests with *march data backgrounds (MDBs)* and *walking data backgrounds (WDBs)*. However, the proposed solution in [5] increases the test time with a factor $d$, where $d$ is the number of DBs. In addition, it does not guarantee the detection of all intra-word faults. In [6, 8], *data-background sequences (DBSs)* for intra-word *idempotent* and *disturb* coupling faults have been presented. In addition, a systematic way for converting BOM tests into WOM tests has been introduced, based on concatenating the BOM test (which detects faults between words) and the WOM test (which detects the faults within a word). However, only idempotent and disturb coupling faults have been considered. As mentioned previously, currently published work shows the existence of many new coupling faults.

This paper presents all possible intra-word coupling faults. The DBSs required for the detection of each of these faults will be established. Thereafter, they will be compiled into a single test detecting all targeted faults. The paper is organized as follows. Section 2 describes the fault models for WOMs; Section 3 establishes the DBS for each intra-word coupling fault; Section 4 introduces a test covering all targeted faults; Section 5 ends with conclusions.

## 2 Fault models for word-oriented memories

Fault models for WOMs can be divided into the *single-cell* faults and *faults between memory cells*.

### 2.1 Single-cell faults

These are faults involving a single cell; they consist of [1, 2, 3, 4, 7]: stuck-at-fault (SAF), transition fault (TF), read destructive fault (RDF), deceptive read destructive fault (DRDF), incorrect read fault (IRF), write disturb fault (WDF), and data retention fault (DRF).

### 2.2 Faults between memory cells

This class consists of coupling faults. There are seven coupling fault models that have been shown to exist in RAMs [2, 4]; where each fault model is defined as a set of *fault primitives (FPs)* [9]. For the FPs, the following notation is used: $< S_a; S_v/F/R >_{a,v}$ (or $< S_a; S_v/F/R >$).

Proceedings of the 21st IEEE VLSI Test Symposium (VTS'03)
1093-0167/03 $17.00 © 2003 IEEE

$S_a$ describes the sensitizing operation or state of the *aggressor cell (a-cell)*; while $S_v$ describes the sensitizing operation or state of the *victim cell (v-cell)*. The a-cell ($c_a$) is the cell sensitizing a fault in another cell, called the v-cell ($c_v$). The set $S_i$ is defined as: $S_i \in \{0, 1, 0w0, 1w1, 0w1, 1w0, r0, r1\}$ ($i \in \{a, v\}$), whereby 0 (1) denotes a *zero* (*one*) value, $0w0$ ($1w1$) denotes a write 0 (1) operation to a cell which contains a 0 (1), $0w1$ ($1w0$) denotes an up (down) transition write operation, and $r0$ ($r1$) denotes a read 0 (1) operation from a cell containing 0 (1).

$F$ describes the value of the *faulty* cell (v-cell) due to a certain $S_a$ and/or $S_v$; $F \in \{0, 1\}$.

$R$ describes the logical value which appears at the output of the memory if the sensitizing operation applied to the v-cell, $S_v$, is a *read* operation: $R \in \{0, 1, -\}$. A '$-$' in $R$ means that the output data is not applicable; e.g., if $S_a = 0$ and $S_v = 1w0$, then no data will appear at the memory output, and for that reason $R$ is replaced by a '$-$'.

As mentioned previously, the coupling faults consist of seven fault models (FMs); they are given below and summarized with their FPs in Table 1.

**Table 1. List of coupling faults**

| FM | # | FP description | # | FP description |
|---|---|---|---|---|
| CFst | 1 | $< 0; 0/1/- >$ | 3 | $< 1; 0/1/- >$ |
| | 2 | $< 0; 1/0/- >$ | 4 | $< 1; 1/0/- >$ |
| CFds | 1 | $< 0w1; 0/1/- >$ | 7 | $< 1w0; 0/1/- >$ |
| | 2 | $< 0w1; 1/0/- >$ | 8 | $< 1w0; 1/0/- >$ |
| | 3 | $< 0w0; 0/1/- >$ | 9 | $< 1w1; 0/1/- >$ |
| | 4 | $< 0w0; 1/0/- >$ | 10 | $< 1w1; 1/0/- >$ |
| | 5 | $< r0; 0/1/- >$ | 11 | $< r1; 0/1/- >$ |
| | 6 | $< r0; 1/0/- >$ | 12 | $< r1; 1/0/- >$ |
| CFtr | 1 | $< 0; 0w1/0/- >$ | 3 | $< 1; 0w1/0/- >$ |
| | 2 | $< 0; 1w0/1/- >$ | 4 | $< 1; 1w0/1/- >$ |
| CFwd | 1 | $< 0; 0w0/1/- >$ | 3 | $< 1; 0w0/1/- >$ |
| | 2 | $< 0; 1w1/0/- >$ | 4 | $< 1; 1w1/0/- >$ |
| CFrd | 1 | $< 0; r0/1/1 >$ | 3 | $< 1; r0/1/1 >$ |
| | 2 | $< 0; r1/0/0 >$ | 4 | $< 1; r1/0/0 >$ |
| CFdr | 1 | $< 0; r0/1/0 >$ | 3 | $< 1; r0/1/0 >$ |
| | 2 | $< 0; r1/0/1 >$ | 4 | $< 1; r1/0/1 >$ |
| CFir | 1 | $< 0; r0/0/1 >$ | 3 | $< 1; 0r0/0/1 >$ |
| | 2 | $< 0; r1/1/0 >$ | 4 | $< 1; 1r1/1/0 >$ |

1. **State coupling fault (CFst)**. The v-cell is forced into a given logic state (e.g., 1) only if the a-cell is in a given state (e.g., 0), without performing any operation on the v-cell or on the a-cell (e.g., $< 1; 1/0/- >$). The CFst consists of four FPs.
2. **Disturb coupling fault (CFds)**: An operation (write or read) performed on the a-cell causes the v-cell to flip. The CFds consists of 12 FPs; see Table 1.
3. **Transition coupling fault (CFtr)**: A given logic value in the a-cell results in a failing transition write operation performed on the v-cell. This fault is sensitized by first setting the a-cell in a given state, and thereafter applying a write operation on the v-cell. The CFtr consists of four FPs; see Table 1.
4. **Write Destructive coupling fault (CFwd)**.
5. **Read Destructive coupling fault (CFrd)**.
6. **Deceptive Read Destructive coupling fault (CFdr)**.
7. **Incorrect Read coupling fault (CFir)**.

The CFs can be further divided, based on the location of the *aggressor cell (a-cell)* and the *victim cell (v-cell)*, into:

1. **Inter-word faults**: These are faults where the a-cell and the v-cell belong to *different* words.
2. **Intra-word faults**: These are faults where the a-cell and the v-cell belong to the *same* word.

Based on the above, one can conclude that the test procedure for WOMs can be divided into three phases:

1. Tests for single-cell faults.
2. Tests for inter-word coupling faults.
3. Tests for intra-word coupling faults.

Testing of single-cell and inter-word coupling faults can be done using BOM tests. The fact that the memory word is $B$ bits wide does not influence the detectability of these faults. Therefore the BOM tests can be converted into WOM tests by replacing the single bit operations $r0$, $r1$, $w0$, and $w1$ with a $B$-bit wide data-background (DB). For instance the $w0$ operation should be replaced with $w_D$, where $D$ can be any DB; for example $w00...0$, $w10...10$, etc. However, the detection of intra-word coupling faults can not be guaranteed with BOM tests. The following example will clarify this. Figure 1 shows a 4-cell memory word ($C_w$, $C_x$, $C_y$, and $C_z$), and the CFdr denoted as $< 0; r1/0/1 >_{w,x}$ (see Table 1); i.e., applying a $r1$ operation to cell $C_x$, while cell $C_w$ is in state 0, will cause cell $C_x$ to flip to 0, while the read operation returns a correct value 1. In order to sensitize the fault, the DB '$01yz$' has to be written ($C_y$ and $C_z$ can have arbitrary values), and then it has to be read. The detection of the fault will require an additional read operation since the first read operation returns the expected value. The fault will thus be detected if the following sequence is applied: '$w01yz$, $r01yz$, $r01yz$', where $y, z \in \{0, 1\}$. Applying any BOM test (with, e.g., DB '0000') will not detect the fault.

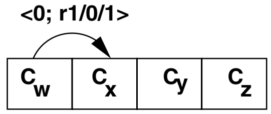

**Figure 1. Intra-word CFdr:** $< 0; r1/0/1 >_{w,x}$

It is important to note here that the intra-word faults will not be detected by BOMs, only when their fault effect dominates the performed operation to the v-cell. E.g., in the example of Figure 1, when the fault effect of CFdr dominates the '$r1$' operation performed to the v-cell.

## 3 DBSs for intra-word coupling faults

Intra-word faults consist of seven coupling faults. This section gives the required DBSs for each of the them.

### 3.1 DBS for intra-word CFds

In the following, the DBS for intra-word CFds will be introduced while considering all possible CFds; i.e., CFds sensitized by either a transition write, a non-transition write, or a read operation.

The CFds consists of 12 FPs; see Table 1. For a $B$-bit WOM , each FP has $C_2^B = \frac{B*(B-1)}{2}$ possible cases since any of the $B$ cells can be the a-cell, while the $B-1$ non a-cells can be the v-cell. The total number of intra-word CFds is therefore $12 * C_2^B = 6 * B * (B-1)$. One has to establish the minimal number of DBSs which sensitize all these faults, and the *minimal* number of operations using these DBSs.

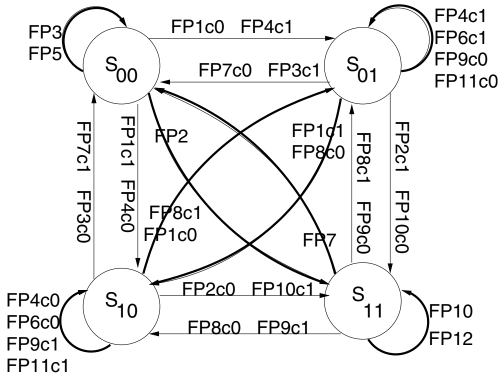

**Figure 2. State diagram for CFds**

Figure 2 shows the state diagram for sensitizing all intra-word CFds within a 2-bit WOM. The states (nodes) are labeled according to the value of the two cells $c_0$ and $c_1$ in the word; the arcs are labeled with the sensitized FPs. In the diagram, e.g., 'FP1c0' denotes FP1 of the CFds shown in Table 1, where the v-cell is $c_0$; i.e., $< 0w1; 0/1/- >_{c1,c0}$. When the FP is specified without 'c0' and without 'c1', then the FP is sensitized in both cases (i.e., when $c_0$ is the v-cell *and* when $c_1$ is the v-cell); e.g., FP2 denotes that $< 0w1; 1/0/- >$ is sensitized for $c_0$ as the v-cell as well as for $c_1$ as the v-cell. By inspecting the diagram, one can conclude that:

- A given FP is sensitized by more than one arc.
- Some FPs are only sensitized by a single arc. These FPs are sensitized by the arcs starting and ending in the same state; e.g., $(S_{01}, S_{01})$ is the only arc sensitizing FP6c1=$< r0; 1/0/- >_{c0,c1}$.
- The arcs starting and ending in the same state sensitize all FPs, except FP1, FP2, FP7 and FP8. The *minimum* set of arcs that can sensitize the latter faults consists of the four arcs connecting states which are each other inverse; i.e., $(S_{00}, S_{11})$ and $(S_{01}, S_{10})$.

Based on the above, on can conclude that in total eight arcs are needed to sensitize all CFds shown as the *bold* arcs of Figure 2. They consist of two subsets of arcs that are disconnected. Hence, an extra arc is required to connect the two subsets; this can be any arc connecting the two subsets. The DBS for CFds can be obtained by concatenating two *sub-data-backgrounds sequences (sDBSs)* $\mathbf{S}^0$= 00, 11, 00; and $\mathbf{S}^1$= 01,10,01; that means that the DBS for a 2-bit word is $\mathbf{S}_2$= 00, 11, 00, 01, 10, 01. Note that two different ways to concatenate $\mathbf{S}^0$ and $\mathbf{S}^1$ exist. In addition, each sDBS can be replaced with its inverse; e.g., $\mathbf{S}^0$= 00, 11, 00 can be replaced with $\overline{\mathbf{S^0}}$= 11, 00, 11, because both $\mathbf{S}^0$ and $\overline{\mathbf{S^0}}$ sensitize the same faults. Therefore the DBS for a 2-bit word $\mathbf{S^2}$ can be constructed in eight different ways: (1) $\mathbf{S}^0$, $\mathbf{S}^1$; (2) $\mathbf{S}^1$, $\mathbf{S}^0$; (3) $\overline{\mathbf{S^0}}$, $\overline{\mathbf{S^1}}$; (4) $\overline{\mathbf{S^1}}$, $\overline{\mathbf{S^0}}$; (5) $\mathbf{S}^0$, $\overline{\mathbf{S^1}}$; (6) $\overline{\mathbf{S^1}}$, $\mathbf{S^0}$; (7) $\overline{\mathbf{S^0}}$, $\mathbf{S}^1$; and (8) $\mathbf{S}^1$, $\overline{\mathbf{S^0}}$.

Consider the DBS for 2-bit words to be $\mathbf{S}_2$= $\mathbf{S}^0$, $\mathbf{S}^1$= 00, 11, 00, 01, 10, 01. Using this DBS, a sequence of read and write operations, $\Omega_2$, can be generated assuming the initial sate 00; this will be: w11,w11,r11,w00,w00,r00,**w01**,w10,w10,r10,w01,w01,r01. These operations are required to *sensitize* all CFds; remember that the CFds can be sensitized with transition write, non-transition write, or read operations. The 'w01' given in bold font is needed only to connect the sDBSs; see Figure 2. To detect the faults using the operation sequence $\Omega_2$, each sensitizing operation has to be followed with a read operation; this read operation will detect the CFds sensitized by the preceding transition write, non-transition write, or read operation. The operation sequence $\Omega_2$ will then be:

w11,*r11*,w11,*r11*,r11, *r11*,w00,*r00*,w00,*r00*,r00,*r00*, **w01**,

w10,*r10*,w10,*r10*,r10, *r10*, w00,*r01*,w01,*r01*,r01,*r01*

The $\Omega_2$ now contains sequences of three identical read operations. The first read operation is required to detect the faults sensitized by the preceding transition write operations; the second read operation is required to detect the faults sensitized by the non-transition write operations; the fourth read operation is required to detect the faults sensitized by the preceding read operation. Hence, the third read operation is redundant and can be removed. The simplified $\Omega_2$ is shown in Figure 3, assuming the initial state 00. The $\Omega_2$ consists of $\Omega_2'$ concatenated with $\Omega_2''$.

| | |
|---|---|
| $\Omega_2'$ | w11, r11, w11, r11, r11, w00, r00, w00, r00, r00, **w01**, |
| $\Omega_2''$ | w10, r10, w10, r10, r10, w01, r01, w01, r01, r01 |

**Figure 3. Operation sequence for CFds**

Extending the 2-bit DBS to a DBS for a $B$-bit word, requires the following steps:

1. *Level 0*: For each cell-pair $(c_i, c_{i+1})$, apply the DBS for 2-bit words $\mathbf{S}_2$= 00, 11, 00, 01, 10, 01. All CFds between $(c_i, c_{i+1+k*2})$ are sensitized, whereby $i+1+k*2$ can be at the most $B-1$; that means that $k \in \{0, 1, ..., \lfloor\frac{(B-1)-(i+1)}{2}\rfloor\}$.
2. *Level 1*: For each cell-pair $(c_i, c_{i+2})$, apply only the sDBS $\mathbf{S}^1$ = 01, 10, 01; this is sufficient because the sDBS $\mathbf{S}^0$ = 00, 11, 00 has already been applied in *Level 0*. All CFds between $(c_i, c_{i+2+k*4})$ are sensitized, where $k \in \{0, 1..., \lfloor\frac{(B-1)-(i+2)}{4}\rfloor\}$.

...

$\lfloor\log_2 B\rfloor$. *Level* $\log_2 B - 1$: For each cell-pair $(c_i, c_{i+2^{Log_2 B-1}})$, apply only the sDBS $\mathbf{S}^1$ = 01, 10, 01 . All CFds between $(c_i, c_{i+2^{Log_2 B-1}+k*2^{Log_2 B}})$ are sensitized, where $k \in \{0, 1..., \lfloor\frac{(B-1)-(i+2^{Log_2 B-1})}{2^{Log_2 B}}\rfloor\}$.

Extension of the operation sequence to detect all CFds in a $B$-bit WOM requires the following steps:

1. *Level 0*: For each cell-pair $(c_i, c_{i+1})$, apply the operation sequence $\Omega_2$ shown in Figure 3. All CFds between $(c_i, c_{i+1+k*2})$ are detected, where $k \in \{0, 1, ..., \lfloor\frac{(B-1)-(i+1)}{2}\rfloor\}$.
2. *Level j*: $(1 \leq j \leq \lfloor\log_2 B\rfloor - 1)$. For each cell-pair $(c_i, c_{i+2^j})$, apply only the sub-operation sequence $\Omega_2''$ preceded with 'w01'; see Figure 3. This is sufficient because the operation sequence $\Omega_2'$ has already been applied in *Level 0*. All CFds between $(c_i, c_{i+2^j+k*2^{j+1}})$ are detected, where $k \in \{0, 1..., \lfloor\frac{(B-1)-(i+2^j)}{2^{j+1}}\rfloor\}$.

As can be seen from the above, the *number of data backgrounds* needed to sensitize all CFds, denoted as $d$, within a word is: $d$ = 6 {the $d$ used for *Level 0*} + 3 {the $d$ used in each additional level} $*(\lfloor(log_2 B\rfloor - 1)$ {the number of additional levels}= $3 + 3 * \lfloor log_2 B\rfloor$. On the other hand the number of operations needed to detect all CFds within a $B$-bit word is: 21 {the number of operations used in *Level 0*} + 11 {the number of operations used in additional levels} $*(\lfloor log_2 B\rfloor - 1)$= 10+11$*\lfloor log_2 B\rfloor$.

## 3.2 DBS for intra-word CFst, CFir, CFrd, and CFdr

In order to detect the CFst, CFir and CFrd between cells in a word, all states of two arbitrary cells should be generated, and in each state a read operation should be applied to the v-cell; each of the two cells can be the v-cell. However, in order to detect CFdr, not *one* read operation has to be applied when the state is generated, but *two*, because the first read operation will *only sensitize* the fault, while the second will *detect* the fault. Therefore, any test detecting CFdr also detects CFst, CFir and CFrd.

The CFdr consists of 4 FPs; see Table 1. For a WOM with $B$-bit words, the total number of intra-word CFdr is $4 * C_2^B = 2 * B * (B-1)$. Figure 4 shows the state diagram for sensitizing all intra-word CFdr within a 2-bit WOM. The states are labeled according to the value of the two cells $c_0$ and $c_1$ in the word, and the arcs are labeled with the sensitized FPs; e.g., FP1=$< 0; r0/1/0 >$ (see Table 1). By inspecting the diagram, one can see that each FP is sensitized by only one arc, which starts and ends in the same state; and that the diagram consists of four disconnected subsets of arcs. Hence, extra arcs are required to connect the four subsets. The DBS for CFdr can be obtained by concatenating the four states: 00, 11, 10, 01. There are $4 * 3 * 2 * 1 = 24$ ways to concatenate these states. Consider the DBS for CFdr for a 2-bit word to be chosen as: $\mathbf{S_2}$= 11, 00, 10, 01.

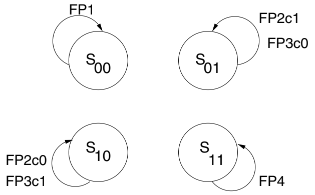

**Figure 4. State diagram for CFdr**

Using $\mathbf{S_2}$= 11, 00, 10, 01, a sequence of read and write operations, $\Omega_2$, can be generated, assuming the initial sate 00; this will be: w11,*r11*,w00,*r00*,w10,*r10*,w01,*r01*. All operations given in italic font are required to *sensitize* all CFdr; the other operations are required to change from one state to another (see Figure 4). To detect the faults using the operation sequence $\Omega_2$, each sensitizing operation has to be followed with a read operation. The operation sequence $\Omega_2$ will then be as shown in Figure 5, assuming the initial state 00; in the figure, the sensitization and the detection operations are printed in italic. The $\Omega_2$ consists of $\Omega_2'$ concatenated with $\Omega_2''$.

| | |
|---|---|
| $\Omega_2'$ | w11,*r11*,*r11*, w00,*r00*,*r00*, |
| $\Omega_2''$ | w10,*r10*,*r10*,w01,*r01*,*r01* |

**Figure 5. Operation sequence for CFdr**

Extending the 2-bit DBS to a DBS for a $B$-bit word, requires the following steps:

1. *Level 0*: For each cell-pair $(c_i, c_{i+1})$, apply the DBS for 2-bit words $\mathbf{S}^2$= 11, 00, 10, 01. The four states between $(c_i, c_{i+1+k*2})$ are generated, where $k \in \{0, 1, ..., \lfloor\frac{(B-1)-(i+1)}{2}\rfloor\}$.
2. *Level j*: $(1 \leq j \leq \lfloor\log_2 B\rfloor - 1)$. For each cell-pair $(c_i, c_{i+2^j})$, apply only the '10, 01'; this is sufficient because the the states 00 and 11 have already been generated in *Level 0*. The states 10 and 01 between $(c_i, c_{i+2^j+k*2^{j+1}})$ are generated , where $k \in \{0, 1..., \lfloor\frac{(B-1)-(i+2^j)}{2^{j+1}}\rfloor\}$.

Extension of the operation sequence to detect all CFdr (also CFst, CFir and CFrd) in a $B$-bit WOM requires the following steps:

1. *Level 0*: For each cell-pair $(c_i, c_{i+1})$, apply the operation sequence shown in Figure 5 (i.e., $\Omega_2'$ and $\Omega_2''$). All CFdr between $(c_i, c_{i+1+k*2})$ are detected, where $k \in \{0, 1, ..., \lfloor\frac{(B-1)-(i+1)}{2}\rfloor\}$.
2. *Level j*: $(1 \leq j \leq \lfloor\log_2 B\rfloor - 1)$. For each cell-pair $(c_i, c_{i+2^j})$, apply only the sub-operation sequence $\Omega_2''$; see Figure 5. This is sufficient because the operation sequence $\Omega_2'$ has already been applied in *Level 0*. All CFdr between $(c_i, c_{i+2^j+k*2^{j+1}})$ are detected, where $k \in \{0, 1..., \lfloor\frac{(B-1)-(i+2^j)}{2^{j+1}}\rfloor\}$.

From the above, it follows that the *number of data backgrounds* needed to sensitize all CFdr, denoted by $d$, within a word is: $d = 4 + 2 * (\lfloor log_2 B\rfloor - 1) = 2 + 2 * \lfloor log_2 B\rfloor$. On the other hand the number of operations needed to detect all CFdr within a $B$-bit word is: $12 + 6 * (\lfloor log_2 B\rfloor - 1) = 6 + 6 * \lfloor log_2 B\rfloor$.

## 3.3 DBS for intra-word CFtr

To detect the CFtr between cells in a word, all states of two arbitrary cells should be generated, and in each state a transition write operation should be applied to the v-cell (while the a-cell should keep its state unchanged) in order to sensitize the fault, followed by a read operation in order to detect the fault.

The CFtr consists of 4 FPs; see Table 1. For a WOM with $B$-bit words, the total number of intra-word CFtr FPs is $4 * C_2^B = 2 * B * (B - 1)$. Figure 6 shows the state diagram for sensitizing all intra-word CFtr within a 2-bit WOM. The arcs are labeled with the sensitized FPs; e.g., FP1c1=$< 0; 0w1/0/- >_{c0,c1}$ (see Table 1). Note that the arcs which connect states which are each others inverse (i.e., $(S_{00}, S_{11})$ and $(S_{01}, S_{10})$) are not included since they do not sensitize any CFtr. By inspecting the diagram, one can see that each FP is sensitized by only one arc, and that the DBS for CFtr can be constructed in two ways (starting at 00):

1) $\mathbf{S}_2$= 01, 11, 10, 00, 10, 11, 01, 00; or
2) $\mathbf{S}_2$= 10, 11, 01, 00, 01, 11, 10, 00.

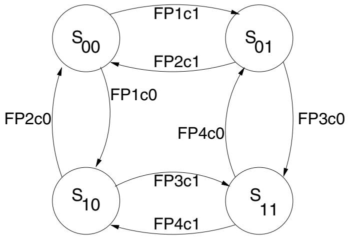

**Figure 6. State diagram for CFtr**

Using $\mathbf{S}_2$= 01, 11, 10, 00, 10, 11, 01, 00, a sequence of sensitizing write operations can be generated; this will be: w01, w11, w10, w00, w10, w11, w01, w00. These operations are required to *sensitize* all CFtr; see Figure 7. To detect the faults, each sensitizing operation has to be followed with a read operation. The operation sequence $\Omega_2$ is shown in Figure 7, assuming the initial state 00.

w01, r01, w11, r11, w10, r10, w00, r00,
w10, r01, w11, r11, w01, r01, w00, r00.

**Figure 7. Operation sequence for CFtr**

In order to extend the 2-bit DBS for a $B$-bit memory, the following steps have to be followed:

1. *Level 0*: For each cell-pair $(c_i, c_{i+1})$, apply the DBS for 2-bit words $\mathbf{S}_2$= 01, 11, 10, 00, 10, 11, 01, 00. All CFtr between $(c_i, c_{i+1+k*2})$ are sensitized, where $i + 1 + k * 2$ can be at the most $B - 1$; that means that $k \in \{0, 1, ..., \lfloor\frac{(B-1)-(i+1)}{2}\rfloor\}$.
2. *Level j*: $(1 \leq j \leq \lfloor\log_2 B\rfloor - 1)$. For each cell-pair $(c_i, c_{i+2^j})$, apply the same $\mathbf{S}_2$= 01, 11, 10, 00, 10, 11, 01, 00. All CFtr between $(c_i, c_{i+2^j+k*2^{j+1}})$ are sensitized, where $k \in \{0, 1..., \lfloor\frac{(B-1)-(i+2^j)}{2^{j+1}}\rfloor\}$.

As can be seen from the above, the *number of data backgrounds* needed to sensitize all CFtr, $d$, within a word is: 8 $\{$the $d$ used for *for each level*$\}$ * $\lfloor\log_2 B\rfloor$ $\{$the number of levels$\}$= $8 * \lfloor\log_2 B\rfloor$. Extension of the operation sequence to detect all CFtr in a $B$-bit WOM can be obtained by adding a read operation after each write operation. Therefore, the number of operations needed to detect all CFtr within a $B$-bit word is: 2$*$ $\{$the number of DBSs$\}$ = $16 * \lfloor\log_2 B\rfloor$.

## 3.4 DBS for intra-word CFwd

To sensitize the CFwd between cells in a word, all states of two arbitrary cells should be generated, and in each state a non-transition write operation should be applied to the v-cell (while the other cell should keep its state unchanged); both of the two cells can be the v-cell. To detect the fault, the write operation has to be followed with a read operation.

The CFwd consists of 4 FPs; see Table 1. For a WOM with $B$-bit words, the total number of intra-word CFwd FPs is $2 * B * (B - 1)$. The state diagram for sensitizing all intra-word CFwd within a 2-bit WOM is exactly the same as the one shown in Figure 4. The arcs are labeled with the sensitized FPs; e.g., FP1=$< 0; 0w0/1/- >$ (see Table 1). The DBS for CFwd can be obtained by concatenating the four states: 00, 11, 10, 01. There are 24 ways to concatenate these states. Consider the DBS for CFwd for a 2-bit word to be: $\mathbf{S_2}$= 11, 00, 10, 01.

Using $\mathbf{S_2}$= 11, 00, 10, 01, a sequence of sensitizing write operations, $\Omega_2$ can be generated; this will be: w11,*w11*,w00,*w00*,w10,*w10*,w01,*w01*. All operations given in italic font are required to *sensitize* all CFwd; the other operations are required to change from one state to another; see Figure 4. To detect the faults, each sensitizing operation has to be followed with a read operation. The operation sequence $\Omega_2$ will be that shown in Figure 8, assuming the initial state 00; in the figure, the sensitization and the detection operations are printed in italic. The $\Omega_2$ consists of $\Omega_2'$ concatenated with $\Omega_2''$.

| | |
|---|---|
| $\Omega_2'$<br>$\Omega_2''$ | w11,*w11*,*r11*, w00,*w00*,*r00*,<br>w10,*w10*,*r10*, w01,*w01*,*r01* |

**Figure 8. Operation sequence for CFwd**

The extension of the 2-bit DBS and the operation sequence for $B$-bit words can be done in the same way as it is has been done for CFdr. The *number of data backgrounds* all CFwd within a word is thus $d = 2+2*\lfloor log_2 B \rfloor$, and the number of operations needed to detect them is $6 + 6 * \lfloor log_2 B \rfloor$.

## 4 WOM march tests for intra-word CFs

Table 2 summarizes the number of DBs, $d$, and the number of operations, $t$, for the intra-word coupling faults of Section 3. It also gives the ration $d/t$ for $B = 8$ and for $B = 16$. By inspecting the table, one can conclude that:

1. The number of operation sequences required for each intra-word CF has a time complexity of $O(log_2 B)$; and therefore the time complexity of the required tests for each CF for a memory with size $n$ is $O(n * log_2 B)$.
2. The CFtr is the fault requiring the largest number of DBs and the largest number of operation sequences.

**Table 2. Number of DBs and of operations**

| Fault | # of DBs $d$ | # of operations $t$ | B=8 | B=16 |
|---|---|---|---|---|
| CFds | $3 + 3 * \lfloor log_2 B \rfloor$ | $10 + 11 * \lfloor log_2 B \rfloor$ | 12/51 | 15/64 |
| CFdr | $2 + 2 * \lfloor log_2 B \rfloor$ | $6 + 6 * \lfloor log_2 B \rfloor$ | 8/24 | 10/30 |
| CFtr | $8 * \lfloor \log_2 B \rfloor$ | $16 * \lfloor \log_2 B \rfloor$ | 24/48 | 32/74 |
| CFwd | $2 + 2 * \lfloor log_2 B \rfloor$ | $6 + 6 * \lfloor log_2 B \rfloor$ | 8/24 | 10/30 |

Since the CFtr is the fault requiring the largest number of DBs, its DBs will be used to establish the required *operation sequence (OS)* to sensitize and detect all intra-word CFs.

Consider the state diagram of CFtr shown in Figure 6. By comparing that diagram with that of CFdr and with that of CFwd shown in Figure 4, one can conclude that the state diagram of CFtr can be extended to cover the arcs required to sensitize the CFdr and CFwd. On the other hand, comparing the CFtr state diagram with that of CFds shown in Figure 2, the reader can easy verify that the CFtr state diagram can also be extended with extra arcs and used to generate the DBS for CFds. Therefore, the CFtr state diagram can be used to establish the DBS and the OS detecting all intra-word CFs (i.e., CFds, CFdr, CFrd, CFir, CFwd, CFtr, CFst). Note that the OS will be optimal for detecting all intra-word CFs, but not optimal for detecting CFds only.

Table 3 shows the DBs and the OS required for each of the intra-word faults for $B = 2$. The first column gives the label of the DB; the second column gives the DBs (i.e., 01,11,10,00,10,11,01,00) as required by CFtr. The third column gives the OS required for CFtr; e.g., in the first row, '$w_{D_0}, r_{D_0}, r_{D_0}$' denotes $w01, r01, r01$; and '$w_{D_1}, r_{D_1}, r_{D_1}$' in the second row denotes $w11, r11, r11$; etc. The third column lists the required OS for CFdr; the reader can verify that only the first four DBs and the corresponding operations are required to sensitize and detect all CFdr; see also Figure 4. A similar explanation applies for fifth column which gives the required OS required for CFwd. The sixth column shows the OS required for CFds; note that only the first four DBs are used to sensitize all CFds based on non-transition write and on read operations, while all DBs are necessary to sensitize all CFds based on transition write operations; see Figure 2. Form Table 3 the following can be concluded:

1. The OS for CFtr is a subset of OS for CFds.
2. The OS for CFdr is a subset of CFds.
3. The OS for CFwd is a subset of CFds, the only difference is that extra read operations are added between the write operation of the OS for CFwd. Adding extra read operations to the OS do not impact the fault coverage.

Therefore the OS for CFds can be used to establish a march test detecting all intra-word faults; it is regiven in column 'all intra-word faults'. The test will have the form: $\{\Updownarrow (operation\ sequences)\}$; where the single march element can be divided in any number of march elements. For $B = 2$, the march test is given in Figure 9; and referred as *March SAM* (**s**ingle-port intr**a**-word **m**emory faults). Note that the total test length is $29 * \frac{n}{B}$, including the initialization.

$$\begin{aligned}
\{\ & \underbrace{\Updownarrow (w00)}_{M_0}\ ; \\
& \underbrace{\Updownarrow (w01, r01, w01, r01, r01, w11, r11, w11, r11, r11)}_{M_1}\ ; \\
& \underbrace{\Updownarrow (w10, r10, w10, r10, r10, w00, r00, w00, r00, r00)}_{M_2}\ ; \\
& \underbrace{\Updownarrow (w10, r10, w11, r11, w01, r01, w00, r00)}_{M_3}\ \}
\end{aligned}$$

**Figure 9. March SAM for B=2**

Establishing march tests for $B$-bit memory can be done by using the following methodology:

1. Extend the DBs for $B$-bit words in the same way as for CFtr; see Section 3.3. The total number of DBs required is $8 * log_2 B$.

Table 3. The DBs and OS for intra-word CFs

| # | DBs | Operation sequences (OS) | | | | |
|---|---|---|---|---|---|---|
| | | CFtr | CFdr | CFwd | CFds | all intra-word faults |
| 0 | 01 | $w_{D_0}, r_{D_0}$ | $w_{D_0}, r_{D_0}, r_{D_0}$ | $w_{D_0}, w_{D_0}, r_{D_0}$ | $w_{D_0}, r_{D_0}, w_{D_0}, r_{D_0}, r_{D_0}$ | $w_{D_0}, r_{D_0}, w_{D_0}, r_{D_0}, r_{D_0}$ |
| 1 | 11 | $w_{D_1}, r_{D_1},$ | $w_{D_1}, r_{D_1}, r_{D_1}$ | $w_{D_1}, w_{D_1}, r_{D_1}$ | $w_{D_1}, r_{D_1}, w_{D_1}, r_{D_1}, r_{D_1}$ | $w_{D_1}, r_{D_1}, w_{D_1}, r_{D_1}, r_{D_1}$ |
| 2 | 10 | $w_{D_2}, r_{D_2}$ | $w_{D_2}, r_{D_2}, r_{D_2}$ | $w_{D_2}, w_{D_2}, r_{D_2}$ | $w_{D_2}, r_{D_2}, w_{D_2}, r_{D_2}, r_{D_2}$ | $w_{D_2}, r_{D_2}, w_{D_2}, r_{D_2}, r_{D_2}$ |
| 3 | 00 | $w_{D_3}, r_{D_3}$ | $w_{D_3}, r_{D_3}, r_{D_3}$ | $w_{D_3}, w_{D_3}, r_{D_3}$ | $w_{D_3}, r_{D_3}, w_{D_3}, r_{D_3}, r_{D_3}$ | $w_{D_3}, r_{D_3}, w_{D_3}, r_{D_3}, r_{D_3}$ |
| 4 | 10 | $w_{D_4}, r_{D_4}$ | $w_{D_4}, r_{D_4}, r_{D_4}$ | $w_{D_4}, w_{D_4}, r_{D_4}$ | $w_{D_4}, r_{D_4}$ | $w_{D_4}, r_{D_4}$ |
| 5 | 11 | $w_{D_5}, r_{D_5}$ | $w_{D_5}, r_{D_5}, r_{D_5}$ | $w_{D_5}, w_{D_5}, r_{D_5}$ | $w_{D_5}, r_{D_5}$ | $w_{D_5}, r_{D_5}$ |
| 6 | 01 | $w_{D_6}, r_{D_6}$ | $w_{D_6}, r_{D_6}, r_{D_6}$ | $w_{D_6}, w_{D_6}, r_{D_6}$ | $w_{D_6}, r_{D_6}$ | $w_{D_6}, r_{D_6}$ |
| 7 | 00 | $w_{D_7}, r_D$ | $w_{D_7}, r_{D_7}, r_{D_7}$ | $w_{D_7}, w_{D_7}, r_{D_7}$ | $w_{D_7}, r_{D_7}$ | $w_{D_7}, r_{D_7}$ |

2. Extend the OS to detect all CFs as follows: for each level $j$ $(1 \leq j \leq \lfloor \log_2 B \rfloor - 1)$, and for each cell-pair $(c_i, c_{i+2^j})$, apply the OS shown in column 'all intra-word faults' in Table 3. All CFs between $(c_i, c_{i+2^j+k*2^{j+1}})$ are sensitized and detected, where $k \in \{0, 1..., \lfloor \frac{(B-1)-(i+2^j)}{2^{j+1}} \rfloor\}$.

   Note that the total number of operations needed to detect all intra-word faults within a $B$-bit word will be: 28 {the number of operation used in each level} $*(\lfloor log_2 B \rfloor)$ {the number of levels}.
3. Write the test as: $\{\Updownarrow (w00..0); \Updownarrow$ (Extended Operation Sequence)}. The $\Updownarrow (w00..0)$ is required for the initialization; the $\Updownarrow$ (Extended Operation Sequence)} can be divided into any number of march elements. The total test length will be: $(1 + 28 * \lfloor log_2 B \rfloor) * \frac{n}{B}$.

### 4.1 Optimization of March SAM

In has been shown above that detecting all intra-word faults in a $B$-bit memory requires a test with a test length of $(1 + 28 * \lfloor log_2 B \rfloor) * \frac{n}{B}$, including the initialization. However, the test length can be reduced to $29 * \frac{n}{B}$, irrespective of the word size $B$ if one assumes intra-word coupling faults to be present only between physically adjacent cells within a word. Each v-cell, say $c_i$, will have at the most two physical neighbors, $c_{i-1}$ and $c_{i+1}$, that can be the a-cell. Therefore, only level 0 of DBs will be required; and only level 0 of the OS will be needed to sensitize and detect all such faults. Generating a March SAM for *restricted* intra-word faults can now be done as follows:

1. For each cell-pair $(c_i, c_{i+1})$, apply the DBS for 2-bit words $\mathbf{S}_2$= 01, 11, 10, 00, 10, 11, 01, 00.
2. For each cell-pair $(c_i, c_{i+1})$, apply the OS shown in column 'all intra-word faults' in Table 3. All CFs between between *physical adjacent* cells within a word are sensitized and detected.
3. Write the test based on OS of step 2. The total test length is $29 * \frac{n}{B}$ including the initialization.

However it is recommend to use $(1 + \frac{28 * \lfloor log_2 B \rfloor}{B}) * n$ March SAM since it has the advantage of also detecting the possible intra-word coupling faults between the different memory buffers.

## 5 Conclusions

In this paper all possible intra-word coupling faults have been presented. The required data-background sequence (DBS) and the operation sequence (OS) to detect each of the coupling faults have been presented. A DBS and OS to detect all intra-word CFs have been presented and compiled into a test detecting all faults. The test, March SAM, has a test length of $(1+28*\lfloor log_2 B \rfloor)*\frac{n}{B}$, including the initialization; where $n$ is the size of the memory and $B$ is the number of bits per word. The test length has also been reduced to $29 * \frac{n}{B}$ when the intra-word faults are restricted to physical adjacent cells within a word.

## References

[1] R.D. Adams and E.S. Cooley, "Analysis of a Deceptive Read Destructive Memory Fault Model and Recommended Testing", *In Proc. IEEE North Atlantic Test Workshop*, 1996.

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

[4] S. Hamdioui, A.J. van de Goor, "Experimental Analysis of Spot Defects in SRAMs: Realistic Fault Models and Tests", *In Proc. of Ninth Asian Test Symposium,* pp. 131-138, 2000.

[5] P.R. Treurer and V.K. Agarwal, "Fault Location Algorithms for Repairable Embedded RAMs", *In Proc. of IEEE Int. Test Conference*, pp. 825-834, 1993.

[6] A.J. van de Goor, I.B.S. Tlili, and S. Hamdioui, "Converting March Tests for Bit-Oriented Memories into Tests for Word-Oriented Memories" *In Proc. of IEEE Int. Workshop on Memory Technology, Design and Testing*, pp. 46-52, 1998.

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

[8] 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*, pp. 501-508, 1998.

[9] 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.

[10] A.J. van de Goor, M.S. Abadir, and A. Carlin, "Minimal Test for Detecting State Coupling Faults Memories", *In Proc. of Design Automation and Test in Europe*, pp. 944-948, 2002.