New Data-Background Sequences and Their Industrial Evaluation for Word-Oriented Random-Access Memories

Said Hamdioui, *Member, IEEE,* and John Eleazar Q. Delos Reyes

***Abstract*—This paper improves upon the state of the art in the testing of intraword coupling faults (CFs) in word-oriented memories. It first presents a complete set of fault models for intraword CFs. Then, it establishes the data background sequence and tests for each intraword CF, as well as a test with complete fault coverage of the targeted faults. All introduced tests will be evaluated industrially, together with the most well-known memory tests. The tests will be applied to big arrays with an interleaved bit-organization as well as to small arrays with an adjacent bit-organization in order to investigate the influence of the memory organization on the intraword CFs. The test results show that the intraword CFs are also significantly important for interleaved memories, even when the cells within a single cell are not physically adjacent. This is due to coupling between the adjacent bit lines and word lines running across the memory array. The paper concludes that intraword CFs should be considered for any serious test purpose or leave substantial defects undetected, especially when considering a high-volume production and a very low defect-per-million (DPM) level.**

***Index Terms*—Bit-oriented memories (BOMs), data backgrounds (DBs), fault models (FMs), memory tests, word-oriented memories (WOMs).**

I. Introduction

RANDOM-ACCESS memories can be organized as bit-oriented memories (BOMs) or as word-oriented memories (WOMs). WOMs contain more than one bit per addressable word, i.e., $B > 2$, whereby B represents the number of bits per word, and B is usually a power of 2. Read operations read B bits simultaneously, while write operations write B data bits simultaneously; where the data to be written in each cell can be specified independently from the data for the other cells.

Traditionally, WOMs have been tested by repeated application of BOM tests, where different data backgrounds (DBs) are used during each iteration [5], [15]–[17], [19]. The disadvantages of this methodology are test time inefficiency and limited fault coverage of coupling faults (CFs) between cells within the same word, which are called intraword CFs. In [5], a systematic method to solve the problem of detecting intraword CFs was 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 [16], a new method based on m-out-of-n codes has been presented to derive an optimal set of DBs. However, the solutions reported in [5] and [16] were restricted only to state CFs. Currently published work shows, based on defect injection and circuit simulation, the existence of many new CFs [3], [7], [10]. In [15], the transformation of BOM tests into WOM tests has been established, based on replacing the data values used by the march tests with march DBs (MDBs) and walking DBs (WDBs). The MDBs replace the data value in the "state changing" march elements, which are elements that consist, in total, of an *odd* number of transition write operations [e.g., $\Uparrow (r0, w1, r1)$], while the WDBs replace the data value in the "state remaining" march elements, which are elements that consist, in total, of an *even* number of transition write operations [e.g., $\Uparrow (r0, w1, r1, w0)$]. However, the solution proposed in [15] increases the test time by a factor d, where d is the number of DBs. In addition, it does not guarantee the detection of all intraword CFs. In [17] and [19], DB sequences (DBSs) for intraword idempotent and disturb CFs 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 CFs have been considered. As mentioned previously, currently published work shows the existence of many new CFs. On the other hand, most used industrial tests repeat the BOM's tests with different DBs, which does not guarantee the detection of all intraword faults. Solid, checkerboard, column stripe, and row stripe are the usual DBs in that case [13], [18].

This paper considers all possible intraword CFs. The DBSs required for the detection of each of these faults will be established, and compiled into tests detecting the targeted faults. The tests will then be evaluated industrially. This paper is organized as follows. Section II describes the fault models (FMs) for WOMs. Section III establishes the DBS and the required operation sequences for each intraword CF. Section IV introduces tests covering the targeted faults. Section V presents the industrial evaluation of the introduced tests and the most well-known memory tests for big arrays with an interleaved bit-organization and for small arrays with an adjacent bit-organization. Section VI ends with the conclusion.

Manuscript received January 7, 2004; revised June 25, 2004. This paper was recommended by Associate Editor S. Hellebrand.

S. Hamdioui is with the Computer Engineering Laboratory, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: Said.Hamdioui@philipscrolles.st.com, S.Hamdioui@ewi.tudelft.nl).

J. E. Q. D. Reyes is with the Intel Corporation, Hillsboro, OR 97124 USA.

Digital Object Identifier 10.1109/TCAD.2005.847904

II. FMs for WOMs

FMs for WOMs can be divided into single-cell faults and faults between memory cells.

A. Single-Cell Faults

These are faults involving a single cell; they consist of [1], [3], [5], [7], [21]: stuck-at fault (SAF), transition fault (TF), read destructive fault (RDF), deceptive RDF (DRDF), incorrect read fault (IRF), write disturb fault (WDF), and data retention fault (DRF).

B. Faults Between Memory Cells

This class consists of CFs. There are seven CF models that have been shown to exist in RAMs [3], [7], [10], where each FM is defined as a set of fault primitives (FPs) [20]. For the FPs, the following notation is used: $\langle S_a; S_v/F/R\rangle_{a,v}$ (or $\langle S_a; S_v/F/R\rangle$).

S_a describes the sensitizing operation or the 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: $S_i \in \{0, 1, 0w0, 1w1, 0w1, 1w0, 0r0, 1r1\}$ $(i \in \{a, v\})$, 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 $0r0$ $(1r1)$ 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 CFs consist of seven FMs; they are given below and summarized with their FPs in Table I.

1) **State CF (CFst)**. Two cells are said to have a state CF if the v-cell is forced into a given logic state, only if the a-cell is in a given state, without performing any operation on the v-cell or on the a-cell. Note that no operation is needed to sensitize the fault; it only depends on the initial stored values in the cells. The CFst consists of four FPs: $\langle 0; 0/1/-\rangle$, $\langle 0; 1/0/-\rangle$, $\langle 1; 0/1/-\rangle$, and $\langle 1; 1/0/-\rangle$.
2) **Disturb CF (CFds)**: Two cells are said to have a disturb CF if an operation (write or read) performed on the a-cell causes the v-cell to flip. Here, any operation performed on the a-cell is accepted as a sensitizing operation for the fault, be it a read, a transition write, or a nontransition write operation. The CFds consist of 12 FPs (see Table I). It should be noted that the known idempotent coupling fault (CFid) [21] is a subset of the CFds; the CFid only consists of the four FPs 1, 2, 7, and 8 (see Table I).
3) **Transition CF (CFtr)**: Two cells are said to have a transition coupling fault if a given logic value in the a-cell results in a failing transition write operation performed on the v-cell. The CFtr consists of four FPs (see Table I).
4) **Write Destructive CF (CFwd)**: Two cells are said to have a write destructive coupling fault if a nontransition write operation performed on the v-cell results in a transition when the a-cell is in a given logic state. The CFwd consists of four FPs (see Table I).
5) **Read Destructive CF (CFrd)**: Two cells are said to have a read destructive coupling fault when a read operation performed on the v-cell changes the data in the v-cell but returns an incorrect value on the output, if the a-cell is in a given state. The CFrd consists of four FPs (see Table I).
6) **Deceptive Read Destructive CF (CFdr)**: Two cells are said to have a deceptive read destructive coupling fault when a read operation performed on the v-cell changes the data in the v-cell and returns a *correct* value on the output, if the a-cell is in a given state. The CFdr consists of four FPs (see Table I).
7) **Incorrect Read CF (CFir)**: Two cells are said to have an incorrect read coupling fault if a read operation performed on the v-cell returns the incorrect logic value when the a-cell is in a given state. Note here that the state of the v-cell is not changed. The CFir consists of four FPs (see Table I).

TABLE I
LIST OF CFs

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	$< 0r0; 0/1/- >$	11	$< 1r1; 0/1/- >$
	6	$< 0r0; 1/0/- >$	12	$< 1r1; 1/0/- >$
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	$< 0; 0r0/1/1 >$	3	$< 1; 0r0/1/1 >$
	2	$< 0; 1r1/0/0 >$	4	$< 1; 1r1/0/0 >$
CFdr	1	$< 0; 0r0/1/0 >$	3	$< 1; 0r0/1/0 >$
	2	$< 0; 1r1/0/1 >$	4	$< 1; 1r1/0/1 >$
CFir	1	$< 0; 0r0/0/1 >$	3	$< 1; 0r0/0/1 >$
	2	$< 0; 1r1/1/0 >$	4	$< 1; 1r1/1/0 >$

The above CFs can be further divided, based on the location of the a-cell and the v-cell, into the following: 1) *Interword faults*, which are faults where the a-cell and the v-cell belong to different words. Classical BOM tests are based on this subclass. 2) *Intraword faults*, which are faults where the a-cell and the v-cell belong to the *same* word. This subclass requires special tests and generally cannot be covered with BOM tests.

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 interword CFs; and 3) tests for intraword CFs.

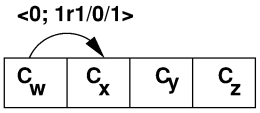

Fig. 1. Intraword CFdr: $\langle 0; 1r1/0/1\rangle_{w,x}$.

TABLE II
OPERATION SEQUENCES FOR INTRAWORD CFS FOR $B = 2$

Fault	Operation sequence Ω_2	
CFds	Ω_2'	w11, r11, w11, r11, r11, w00, r00, w00, r00, r00, **w01**,
	Ω_2''	w10, r10, w10, r10, r10, w01, r01, w01, r01, r01
CFdr	Ω_2'	w11,*r11*,*r11*, w00,*r00*,*r00*,
	Ω_2''	w10,*r10*,*r10*,w01,*r01*,*r01*
CFtr		w01, r01, w11, r11, w10, r10, w00, r00, w10, r10, w11, r11, w01, r01, w00, r00.
CFwd	Ω_2'	w11,*w11*,*r11*, w00,*w00*,*r00*,
	Ω_2''	w10,*w10*,*r10*, w01,*w01*,*r01*

Testing of single-cell and interword CFs 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 $r0$ (i.e., read 0), $r1$, $w0$ (i.e., write 0), and $w1$ with a B-bit-wide DB. For instance, the $w0$ operation should be replaced with w_D, where D can be any DB; for example, $w00\ldots0, w10\ldots10$, etc. However, the detection of intraword CFs cannot be guaranteed with BOM tests. The following example will clarify this. Fig. 1 shows a four-cell memory word (C_w, C_x, C_y, and C_z), and the CFdr denoted as $\langle 0; 1r1/0/1\rangle_{w,x}$ (see Table I); i.e., applying a $r1$ operation to cell C_x containing 1, 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.

It is important to note that the BOM tests may not detect intraword CFs *only* when their fault effect dominates the performed operation to the v-cell. For example, in the example of Fig. 1, when the fault effect of CFdr dominates the "$r1$" operation performed to the v-cell.

III. DBSs FOR INTRAWORD CFs

Intraword CFs consist of seven CFs. This section gives the required DBSs together with the required operation sequences to detect each of the them; the results are given in Table II.

A. DBS for Intraword CFds

In [19], the DBS for the intraword CFds has been introduced. However, only CFds sensitized by transition write operations and read operations have been considered. In the following, the DBS for intraword CFds will be reintroduced while considering all possible CFds, i.e., CFds sensitized by either a transition write, a nontransition write, or a read operation.

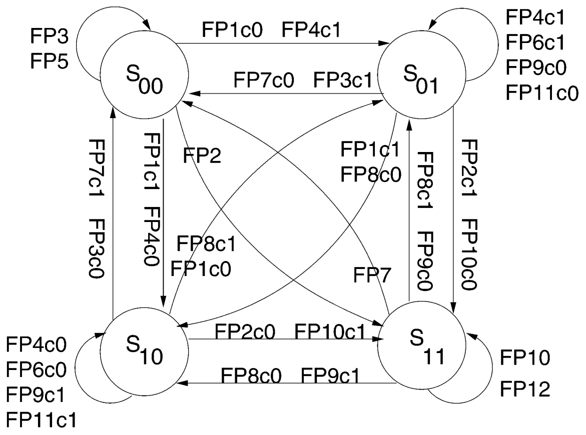

Fig. 2. State diagram for CFds.

The CFds consists of 12 FPs (see Table I). For a B-bit WOM, each FP has $C_2^B = (B*(B-1))/(2)$ possible cases since any of the B cells can be the a-cell, while any of the $B-1$ non a-cells can be the v-cell. The total number of intraword 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.

Fig. 2 shows the state diagram for sensitizing all intraword 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 I, where the v-cell is c_0 (and the a-cell is the c_1); i.e., $\langle 0w1; 0/1/-\rangle_{c1,c0}$. 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 $\langle 0w1; 1/0/-\rangle$ 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 the following.

1) A given FP is sensitized by more than one arc; e.g., $\mathrm{FP1c0} = \langle 0w1; 0/1/-\rangle_{c1,c0}$ is sensitized by the arcs of (S_{00}, S_{01}) and (S_{10}, S_{01}).
2) 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 $\mathrm{FP6c1} = \langle 0r0; 1/0/-\rangle_{c0,c1}$.
3) 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 the states which are each others inverse, i.e., (S_{00}, S_{11}) and (S_{01}, S_{10}).

Based on the above, one can conclude that in total eight arcs are needed to sensitize all CFds; these arcs are given in Fig. 3. Note that the figure consists 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 states in the two subsets. The DBS for CFds can be obtained by concatenating two sub-DBSs (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 complement; 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

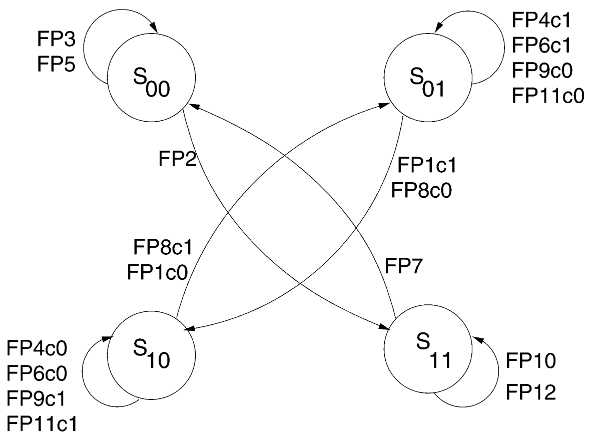

Fig. 3. Minimum arcs sensitizing all CFds.

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 Ω_2 can be generated assuming the initial state 00; this will be: $w11, w11, r11, w00, w00, r00, \mathbf{w01}, w10, w10, r10, w01, w01, r01$. These operations are required to *sensitize* all CFds; remember that the CFds can be sensitized with transition write, nontransition write, or read operations. The emboldened **w01** is needed only to connect the sDBSs (see Fig. 3). To detect the faults using the operation sequence Ω_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, nontransition write, or read operation. The operation sequence Ω_2 will then be

$$w11, r11, w11, r11, r11, r11, w00, r00, w00, r00, r00, r00, \mathbf{w01}, w10, r10, w10, r10, r10, r10, w01, r01, w01, r01, r01, r01.$$

The Ω_2 now contains sequences with 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 nontransition 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 Ω_2 is shown in Table II, assuming the initial state 00. The Ω_2 consists of Ω_2' concatenated with Ω_2''.

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 $k \in \{0, 1, \ldots, \lfloor((B-1)-(i+1))/(2)\rfloor\}$.
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 $k \in \{0, 1 \ldots, \lfloor((B-1)-(i+2))/(4)\rfloor\}$.

...

$\lfloor\log_2 B\rfloor$) *Level* $\log_2 B - 1$: For each cell-pair $(c_i, c_{i+2^{\mathrm{Log}_2 B-1}})$, apply only the sDBS $\mathbf{S}^1 = 01, 10, 01$. All CFds between $(c_i, c_{i+2^{\mathrm{Log}_2 B-1}+k*2^{\mathrm{Log}_2 B}})$ are sensitized, where $k \in \{0, 1 \ldots, \lfloor((B-1)-(i+2^{\mathrm{Log}_2 B-1}))/(2^{\mathrm{Log}_2 B})\rfloor\}$.

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 Ω_2 shown in Table II. All CFds between $(c_i, c_{i+1+k*2})$ are detected, where $k \in \{0, 1, \ldots, \lfloor((B-1)-(i+1))/(2)\rfloor\}$.
2) *Level 1*: For each cell-pair (c_i, c_{i+2}), apply only the suboperation sequence Ω_2'' preceded with "w01" (see Table II). This is sufficient because the operation sequence Ω_2' has already been applied in *Level 0*. All CFds between $(c_i, c_{i+2+k*4})$ are detected, where $k \in \{0, 1 \ldots, \lfloor((B-1)-(i+2))/(4)\rfloor\}$.

...

$\lfloor\log_2 B\rfloor$) *Level* $\log_2 B - 1$: For each cell-pair $(c_i, c_{i+2^{\mathrm{Log}_2(B-1)}})$, apply only "w01" and the suboperation sequence Ω_2'' (see Table II). All CFds between $(c_i, c_{i+2^{\mathrm{Log}_2 B-1}+k*2^{\mathrm{Log}_2 B})})$ are sensitized, where $k \in \{0, 1 \ldots, \lfloor((B-1)-(i+2^{\mathrm{Log}_2 B-1}))/(2^{\mathrm{Log}_2 B})\rfloor\}$.

As can be seen from the above, the *number* of DBs 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} $*$ $(\lfloor\log_2 B\rfloor - 1) = 10 + 11 * \lfloor\log_2 B\rfloor$.

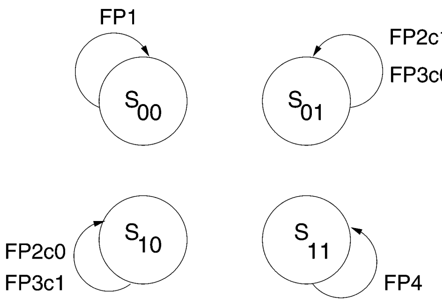

Fig. 4. State diagram for CFdr.

B. DBS for Intraword CFdr

In order to sensitize the CFdr 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, an extra read operation has to be applied. This is because when the state is generated, the first read operation will *only sensitize* the fault, while the second will *detect* the fault.

The CFdr consists of 4 FPs (see Table I). For a WOM with B-bit word, each FP has $C_2^B = (B*(B-1))/(2)$ possible cases; the total number of intraword CFdr is therefore $2*B*(B-1)$. Fig. 4 shows the state diagram for sensitizing all intraword 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., $\mathrm{FP1} = \langle 0; 0r0/1/0\rangle$ (see Table I). By inspecting the diagram, one can see that each FP is sensitized by only one arc, which starts and ends in the same state; one can also see 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, 01, 10, 11. 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: $\mathbf{S_2} = 11, 00, 10, 01$.

Using $\mathbf{S_2} = 11, 00, 10, 01$, a sequence of read and write operations Ω_2 can be generated, assuming the initial state 00; this will be $\mathrm{w}11, r11, \mathrm{w}00, r00, \mathrm{w}10, r10, \mathrm{w}01, r01$. All italicized operations are required to *sensitize* all CFdr; the other operations are required to change from one state to another. To detect the faults using the operation sequence Ω_2, each sensitizing operation has to be followed with a read operation. The operation sequence Ω_2 will then be as shown in Table II, assuming the initial state 00; in the figure, the sensitization and the detection operations are printed in italics. The Ω_2 consists of Ω_2' concatenated with Ω_2''.

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, whereby $i + 1 + k * 2$ can be at the most $B - 1$; which means that $k \in \{0, 1, \ldots, \lfloor((B-1) - (i+1))/(2)\rfloor\}$.
2) [*Level j*: $(1 \le j \le \lfloor\log_2 B\rfloor - 1)$. 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 $k \in \{0, 1 \ldots, \lfloor((B-1) - (i+2^j))/(2^{j+1})\rfloor\}$.

Extension of the operation sequence to detect all CFdr 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 Table II (i.e., Ω_2' and Ω_2''). All CFdr between $(c_i, c_{i+1+k*2})$ are detected, where $k \in \{0, 1, \ldots, \lfloor((B-1) - (i+1))/(2)\rfloor\}$.
2) *Level* j: $(1 \le j \le \lfloor\log_2 B\rfloor - 1)$. For each cell-pair (c_i, c_{i+2^j}), apply only the suboperation sequence Ω_2''; see Table II. This is sufficient because the operation sequence Ω_2' has already been applied in *Level 0*. All CFdr between $(c_i, c_{i+2^j+k*2^{j+1}})$ are detected, where $k \in \{0, 1 \ldots, \lfloor((B-1) - (i+2^j))/(2^{j+1})\rfloor\}$.

From the above, it follows that the *number* of DBs needed to sensitize all CFdr, denoted by d, within a word is: $d = 4$ {the d used for *Level 0*} $+ 2$ {the d used in each additional level} $*$ $(\lfloor(\log_2 B\rfloor - 1)$ {the number of additional levels} $= 2 + 2 * \lfloor\log_2 B\rfloor$. On the other hand, the number of operations needed to detect all CFdr within a B-bit word is: 12 {the number of operations used in *Level 0*} $+$ 6{the number of operations used in additional levels} $*(\lfloor\log_2 B\rfloor - 1) = 6 + 6 * \lfloor\log_2 B\rfloor$.

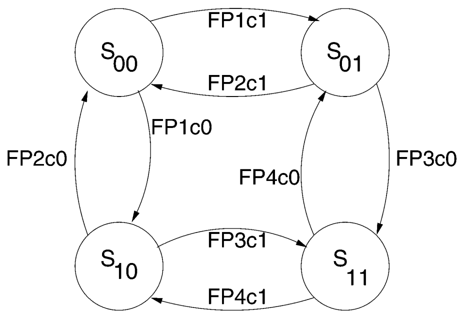

Fig. 5. State diagram for CFtr.

C. DBS for Intraword CFtr

In order 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; each of the two cells can be the v-cell.

The CFtr consists of 4 FPs (see Table I). For a WOM with B-bit words, the total number of intraword CFtr FPs is therefore $2 * B * (B - 1)$. Fig. 5 shows the state diagram for sensitizing all intraword CFtr 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., $\mathrm{FP1c1} = \langle 0; 0w1/0/-\rangle_{c0,c1}$ (see Table I). Note that the arcs which connect states which are each others's inverse (i.e., (S_{00}, S_{11}) and (S_{01}, S_{10})) are not included since they do not sensitize any CFtr because then both cells would undergo a transition write, while the CFtr requires that only *one* cell undergoes a transition write while the other cell should keep its specified state. 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 one of two ways (starting at S_{00}):

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

Using $\mathbf{S}_2 = 01, 11, 10, 00, 10, 11, 01, 00$, a sequence of sensitizing write operations can be generated assuming the initial state 00; this will be $w01, w11, w10, w00, w10, w11, w01, w00$. These operations are required to *sensitize* all CFtr. To detect the faults, each sensitizing operation has to be followed with a read operation. The operation sequence Ω_2 is shown in Fig. 2, assuming the initial state 00.

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 $B - 1$, which means that $k \in \{0, 1, \ldots, \lfloor((B-1) - (i+1))/(2)\rfloor\}$.
2) *Level* j: $(1 \le j \le \lfloor\log_2 B\rfloor - 1)$. 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 $k \in \{0, 1 \ldots, \lfloor((B-1) - (i+2^j))/(2^{j+1})\rfloor\}$.

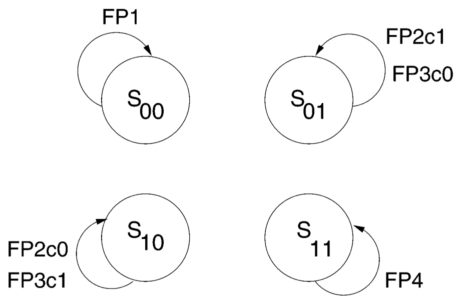

Fig. 6. State diagram for CFwd.

As can be seen from the above, the number d of DBs needed to sensitize all CFtr, 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, since each write operation in the DBS is a sensitizing operation. Therefore, the number of operations needed to detect all CFtr within a B-bit word is $2 * \{$the number of DBSs$\} = 16 * \lfloor \log_2 B \rfloor$.

D. DBS for Intraword CFwd

In order to sensitize the CFwd between cells in a word, all states of two arbitrary cells should be generated, and in each state, a nontransition 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. The write operation has to be followed with a read operation in order to detect the fault.

The CFwd consists of four FPs (see Table I). For a WOM with B-bit words, the total number of intraword CFwd FPs is therefore $2 * B * (B - 1)$. Fig. 6 shows the state diagram for sensitizing all intraword CFwd 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., $\mathrm{FP1} = \langle 0; 0w0/1/- \rangle$ (see Table I). By inspecting the diagram, one can see that each FP is sensitized by only one arc, which starts and ends in the same state; note also that the diagram consists of four disconnected subsets of arcs. Hence, extra arcs are required to connect the four subsets. The DBS for CFwd can be obtained by concatenating the four states: 00, 01, 10, 11. 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 Ω_2 can be generated assuming the initial state 00; this will be w11, $w11$, w00, $w00$, w10, $w10$, w01, $w01$. All italicized operations are required to *sensitize* all CFwd; the other operations are required to change from one state to another (see Fig. 6). To detect the faults using the operation sequence Ω_2, each sensitizing operation has to be followed with a read operation. The operation sequence Ω_2 will be that shown in Table II, assuming the initial state 00; in the figure, the sensitization and the detection operations are printed in italics font. The Ω_2 consists of Ω_2' concatenated with Ω_2''.

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 and, therefore, the number of DBs needed to sensitize all CFwd within a word is the same as that required for CFdr; that is $d = 4$ $\{$the d used for *Level 0*$\}$ + 2 $\{$the d used in each additional level$\}$ $*(\lfloor (\log_2 B \rfloor - 1)$ $\{$the number of additional levels$\} = 2 + 2 * \lfloor \log_2 B \rfloor$. On the other hand, the number of operations needed to detect all CFwd within a B-bit word will be 12 $\{$the number of operations used in *Level 0*$\}$ + 6$\{$the number of operations used in additional levels$\} * (\lfloor \log_2 B \rfloor - 1) = 6 + 6 * \lfloor \log_2 B \rfloor$.

TABLE III
NUMBER OF DBs AND 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

E. DBS for Intraword CFst, CFir, and CFrd

In order to detect the CFst, CFir, and CFrd faults between all 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. By inspecting the state diagrams of each of the previous discussed intraword CFs (i.e., CFds, CFdr, CFwd, and CFtr), one can see that by all these diagrams the four states are generated and that by each operation sequence the read operations are applied to the two cells. Therefore, any test detecting CFds, CFdr, CFwd of CFtr also detects CFst, CFir, and CFrd. The optimal test detecting the intraword CFst, CFir, and CFrd requires the same operation sequence as those of CFdr (see Table II), but with *only* one read operation after each write.

IV. WOM March Tests for Intraword CFs

Based on the operation sequences established in the previous section, a march test can be constructed for each of the intraword faults. The test has the form

$$\{\Updownarrow (w00..0); \Updownarrow (\text{Extended Operation Sequence})\}.$$

The $\Updownarrow$ $(w00..0)$ is required for the initialization; the $\Updownarrow$ (Extended Operation Sequence)$\}$ can be divided into any number of march elements. As an example for $B = 2$, the test for CFdr referred to as "Test CFdr" will be (see also Table II): $\{\Updownarrow (w00); \Updownarrow (w11, r11, r11, w00, r00, r00); \Updownarrow (w10, r10, r10, w01, r01, r01)\}$.

In a similar way, "Test CFds," "Test CFwd," and "Test CFtr" can be constructed. Note that the test length (TL) of each test can be calculated by multiplying the number of operations (see Table III) by (n/B), where n is the size of the memory (i.e., $TL = (1 + t) * (n/B)$); 1 is added for the initialization.

Table III summarizes the number of DBs, d, and the number of operations t for the intraword CFs of Section III. It also gives the ration d/t for $B = 8$ and for $B = 16$. By inspecting Table III, one can conclude the following.

1. The number of operation sequences required for each intraword 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 IV
LIST OF DBs AND OSs FOR INTRAWORD 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}	w_{D_4}, r_{D_4}
5	11	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}	w_{D_6}, r_{D_6}
7	00	w_{D_7}, r_{D_7}			w_{D_7}, r_{D_7}	w_{D_7}, r_{D_7}

A. *Test for All Intraword CFs*

In the following, a test detecting all intraword CFs will be generated. 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 targeted intraword CFs in this paper.

Consider the state diagram of CFtr shown in Fig. 5. By comparing that diagram with that of CFdr shown in Fig. 4 and with that of CFwd shown in Fig. 6, one can conclude that the state diagram of CFtr can be extended to cover the arcs required to sensitize the CFdr, and CFwd. That means that the state diagram can be used to generate the required DBs and OSs detecting all CFtr, CFdr, and CFwd.

On the other hand, comparing the CFtr state diagram with that of CFds shown in Fig. 2, the reader can easily verify that the CFtr state diagram can also be extended with extra arcs to generate the OS for CFds. Therefore, the CFtr state diagram can be used to establish the OS detecting all intraword CFs (i.e., CFds, CFdr, CFrd, CFir, CFwd, CFtr, CFst). However, the OS will not be the optimal one for CFds since the reduced state diagram generating the minimum OS shown in Fig. 3 is different from that of CFtr.

Table IV shows the DBs and the OSs required for each of the intraword CFs 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). The third column gives the OS required for CFtr; e.g., in the first row, "w_{D_0}, r_{D_0}" denotes $w01, r01$; and "w_{D_1}, r_{D_1}" in the second row denotes $w11, r11$; 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 Fig. 4. A similar explanation applies to the 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 nontransition write and on read operations, while all DBs are necessary to sensitize all CFds based on transition write operations; see Fig. 2. From Table IV, it can be concluded that: 1) the OS for CFtr is a subset of OS for CFds; 2) the OS for CFdr is a subset of CFds; and 3) the OS for CFwd is a subset of CFds, the only difference is that extra read operations are added between the write operations of the OS for CFwd. Adding extra read operations to the OS does not negatively impact the fault coverage.

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

Fig. 7. March SAM for $B = 2$.

Therefore, the OS for CFds can be used to establish a march test detecting all intraword faults; it is given in column "all intraword CFs" of Table IV. 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 Fig. 7; and is referred to as *March SAM* (single-port intra-word memory faults). Note that the total test length is $29 * (n/2)$, including the initialization.

Establishing march tests for B-bit memories 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 III-C). The total number of DBs required is $8 * \log_2 B$.
2) Extend the OS to detect all intra-CFs as follows. For each level j $(0 \leq j \leq \lfloor \log_2 B \rfloor - 1)$, and for each cell-pair (c_i, c_{i+2^j}), apply the OS shown in column "all intraword CFs" of Table IV. All intraword CFs between $(c_i, c_{i+2^j+k*2^{j+1}})$ are sensitized and detected, where $k \in \{0, 1 \ldots, \lfloor((B-1)-(i+2^j))/(2^{j+1})\rfloor\}$. Note that the total number of operations needed to detect all intraword CFs 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: $\{\Updownarrow$ $(w00\ldots0); \Updownarrow$ (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 $(1 + 28 * \lfloor \log_2 B \rfloor) * (n/B)$.

The example below illustrates how a march test for all intraword CFs for B-bit word memory can be generated for $B = 4$. Table V shows the DBs and OSs required to detect all intraword CFs ("L" in the table denotes the level). The table is generated using the methodology explained above. Based on the table, March SAM for $B = 4$ can be constructed (see Fig. 8); it has a test length of $57 * (n/4)$, including the initialization.

TABLE V
DBs AND THE OSs FOR $B = 4$

#	L	State $c_0c_1c_2c_3$	Operations
0	0	0 1 0 1	$\mathrm{w}_{D_0}, \mathrm{r}_{D_0}, \mathrm{w}_{D_0}, \mathrm{r}_{D_0}, \mathrm{r}_{D_0}$
1	0	1 1 1 1	$\mathrm{w}_{D_1}, \mathrm{r}_{D_1}, \mathrm{w}_{D_1}, \mathrm{r}_{D_1}, \mathrm{r}_{D_1}$
2	0	1 0 1 0	$\mathrm{w}_{D_2}, \mathrm{r}_{D_2}, \mathrm{w}_{D_2}, \mathrm{r}_{D_2}, \mathrm{r}_{D_2}$
3	0	0 0 0 0	$\mathrm{w}_{D_3}, \mathrm{r}_{D_3}, \mathrm{w}_{D_2}, \mathrm{r}_{D_3}, \mathrm{r}_{D_3}$
4	0	1 0 1 0	$\mathrm{w}_{D_4}, \mathrm{r}_{D_4}$
5	0	1 1 1 1	$\mathrm{w}_{D_5}, \mathrm{r}_{D_5}$
6	0	0 1 0 1	$\mathrm{w}_{D_6}, \mathrm{r}_{D_6}$
7	0	0 0 0 0	$\mathrm{w}_{D_7}, \mathrm{r}_{D_7}$
8	1	0 0 1 1	$\mathrm{w}_{D_8}, \mathrm{r}_{D_8}, \mathrm{w}_{D_8}, \mathrm{r}_{D_8}, \mathrm{r}_{D_8}$
9	1	1 1 1 1	$\mathrm{w}_{D_9}, \mathrm{r}_{D_9}, \mathrm{w}_{D_9}, \mathrm{r}_{D_9}, \mathrm{r}_{D_9}$
10	1	1 1 0 0	$\mathrm{w}_{D_{10}}, \mathrm{r}_{D_{10}}, \mathrm{w}_{D_{10}}, \mathrm{r}_{D_{10}}, \mathrm{r}_{D_{10}}$
11	1	0 0 0 0	$\mathrm{w}_{D_{11}}, \mathrm{r}_{D_{11}}, \mathrm{w}_{D_{11}}, \mathrm{r}_{D_{11}}, \mathrm{r}_{D_{11}}$
12	1	1 1 0 0	$\mathrm{w}_{D_{12}}, \mathrm{r}_{D_{12}}$
13	1	1 1 1 1	$\mathrm{w}_{D_{13}}, \mathrm{r}_{D_{13}}$
14	1	0 0 1 1	$\mathrm{w}_{D_{14}}, \mathrm{r}_{D_{14}}$
15	1	0 0 0 0	$\mathrm{w}_{D_{15}}, \mathrm{r}_{D_{15}}$

$$
\begin{aligned}
&\{\ \underset{M_0}{\Updownarrow (w0000)}\ ;\\
&\underset{M_1}{\Updownarrow (w0101, r0101, w0101, r0101, r0101, w1111, r1111, w1111, r1111, r1111)}\ ;\\
&\underset{M_2}{\Updownarrow (w1010, r1010, w1010, r1010, r1010, w0000, r0000, w0000, r0000, r0000)}\ ;\\
&\underset{M_3}{\Updownarrow (w1010, r1010, w1111, r1111, w0101, r0101, w0000, r0000)}\ ;\\
&\underset{M_4}{\Updownarrow (w0011, r0011, w0011, r0011, r0011, w1111, r1111, w1111, r1111, r1111)}\ ;\\
&\underset{M_5}{\Updownarrow (w1100, r1100, w1100, r1100, r1100, w0000, r0000, w0000, r0000, r0000)}\ ;\\
&\underset{M_6}{\Updownarrow (w1100, r1100, w1111, r1111, w0011, r0011, w0000, r0000)}\ \}
\end{aligned}
$$

Fig. 8. March SAM for $B = 4$.

B. Optimization of March SAM

It has been shown in the previous section that detecting all intraword CFs in a B-bit memory requires a test with a test length of $(1 + 28 * \lfloor \log_2 B \rfloor) * (n/B)$, including the initialization. However, the test length can be reduced to $29 * (n/B)$, irrespective of the word size B, if one assumes intraword CFs 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 OSs will be needed to sensitize and detect all such faults. Generating March SAM for *restricted* intraword CFs 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 intraword CFs" in Table IV. All intraword CFs between *physically adjacent* cells within a word are sensitized and detected.
3. Write the test based on OS of step 2. The total test length is $29 * (n/B)$ including the initialization.

However, it is recommended that one uses $(1 + 28 * \lfloor \log_2 B \rfloor) * (n/B)$ March SAM since it has the advantage of also detecting the possible intraword CFs between the different memory buffers and I/O data paths.

TABLE VI
SUMMARY OF INTRAWORD TESTS

Test	Intra-word fault coverage						
	CFds	CFst	CFir	CFrd	CFdr	CFwd	CFtr
Test CFds	+	+	+	+	+	+	-
Test CFdr	-	+	+	+	+	-	-
Test CFwd	-	+	+	+	-	+	-
Test CFtr	-	+	+	+	-	-	+
March SAM	+	+	+	+	+	+	+

C. Summary of the Introduced Tests

Table VI summarizes the introduced tests for intraword CFs together with their fault coverage. In the table, "+" denotes that a test does detect the corresponding intraword CF and "−" denotes the test does not. E.g., "Test CFdr" detects CFst but not CFds. The evaluation of each tests is done based on the DBs and the operation sequences required to detect each fault. The table shows that all tests detect CFst, CFir, and CFrd; and that "Test CFds" detects all considered faults except CFtr.

D. Influence of the Memory Organization

Generally, a memory chip with a size of n bits, denoted by M_n, consists of s identical two-dimensional subarrays of memory cells. Each subarray M_m contains m bits; $m = q * r$, where q is the number of columns and r the number of rows (i.e., $n = s * m = s * q * r$). Multiple subarrays are used instead of one single array to shorten the word and bit lines and thereby reduce the access time. WOMs can be organized internally in many different ways (depending on where the B bits of a word are physically located within a row of the memory cell array) [17], [21]:

1) *Adjacent*: a q-bit row in a subarray contains $w * B$ bits. The B bits of a word are adjacent.
2) *Interleaved*: a q-bit row in a subarray contains $w * B$ bits. The B bits of a word are spread across B groups in such way that the bits of a B-bit word are interleaved with $w - 1$ bits of the other B-bit words in that row.
3) *Subarrays*: each bit of a B-bit word is taken from a different subarray such that each of the B bits has the same address in each subarray.

Below, the consequences for intraword tests is analyzed for each of the memory organizations.

1) *Adjacent*: the proposed WOM test(s) for intraword CFs have to be fully applied.
2) *Interleaved*: the bits of a B-bit words are separated with the $w-1$ bits of other B-bit words. Therefore, there are no neighboring cells belonging to the same word. As a consequence, no intraword CFs in the memory cell array are possible. However, one has to check, using WOM tests, for the possible intraword CFs that may occur between input/output (I/O) data paths.
3) *Subarrays*: where each of the B bits of an external word is located in a different subarray. BOM tests for CFs will detect the CFs within a B-bit word such that no intraword tests are required. Similar to the interleaved case, one has to verify the possible intraword faults between I/O data paths.

TABLE VII
LIST OF THE USED INTER–BTs

#	BT name	Description
1	SCAN [1]	$\{\Uparrow (w0); \Uparrow (r0); \Uparrow (w1); \Uparrow (r0)\}$
2	MATS+ [12]	$\{\Updownarrow (w0); \Uparrow (r0, w1); \Downarrow (r1, w0)\}$
3	MATS++ [4]	$\{\Updownarrow (w0); \Uparrow (r0, w1); \Downarrow (r1, w0, r0)\}$
4	March C- [11,21]	$\{\Updownarrow (w0); \Uparrow (r0, w1); \Uparrow (r1, w0); \Downarrow (r0, w1); \Downarrow (r1, w0); \Updownarrow (r0)\}$
5	PMOVI [6]	$\{\Downarrow (w0); \Uparrow (r0, w1, r1); \Uparrow (r1, w0, r0); \Downarrow (r0, w1, r1); \Downarrow (r1, w0, r0)\}$
6	March SR [7,10]	$\{\Downarrow (w0); \Uparrow (r0, w1, r1, w0);\Uparrow (r0, r0); \Uparrow (w1); \Downarrow (r1, w0, r0, w1); \Downarrow (r1, r1)\}$
7	March SS [9,10]	$\{\Updownarrow (w0); \Uparrow (r0, r0, w0, r0, w1); \Uparrow (r1, r1, w1, r1, w0); \Downarrow (r0, r0, w0, r0, w1); \Downarrow (r1, r1, w1, r1, w0); \Updownarrow (r0)\}$
8	March G [14]	$\{\Updownarrow (w0); \Uparrow (r0, w1, r1, w0, r0, w1);\Uparrow (r1, w0, w1); \Downarrow (r1, w0, w1, w0); \Downarrow (r0, w1, w0); \Uparrow (r0, w1, r1); \Uparrow (r1, w0, r0)\}$
9	March RAW [8]	$\{\Updownarrow (w0); \Uparrow (r0, w0, r0, r0, w1, r1); \Uparrow (r1, w1, r1, r1, w0, r0); \Downarrow (r0, w0, r0, r0, w1, r1); \Downarrow (r1, w1, r1, r1, w0, r0);\Updownarrow (r0)\}$
10	Hammer [18]	$\{\Uparrow (w0); \Uparrow (r0, 10 * w1, r1); \Uparrow (r1, 10 * w0, r0); \Downarrow (r0, 10 * w1, r1); \Downarrow (r1, 10 * w0, r0)\}$
11	GalColumn [4]	$\{\Uparrow (w0); \Uparrow_b (w1_b, col(r0, r1_b), w0_b); \Uparrow (w1); \Uparrow_b (w0_b, col(r1, r0_b), w1_b)\}$
12	GalRow [4]	$\{\Uparrow (w0); \Uparrow_b (w1_b, row(r0, r1_b), w0_b);\Uparrow (w1); \Uparrow_b (w0_b, row(r1, r0_b), w1_b)\}$
13	WalkColumn [4]	$\{\Uparrow (w0); \Uparrow_b (w1_b, col(r0), r1_b, w0_b); \Uparrow (w1); \Uparrow_b (w0_b, col(r1), r1_b, w0_b)\}$
14	WalkRow [4]	$\{\Uparrow (w0); \Uparrow_b (w1_b, row(r0), r1_b, w0_b); \Uparrow (w1); \Uparrow_b (w0_b, row(r1), r1_b, w0_b)\}$

V. INDUSTRIAL EVALUATION

This section describes an industrial evaluation of the most known memory tests and the tests introduced in the previous section. The DPM screening experiment was done at Intel for advanced embedded caches with big size (which have an interleaved bit-organization) as well as with small size (which have an adjacent bit-organization). The layout of the two types of memory arrays are similar; they differ only in the organization. In addition, they both have sense amplifiers. First, the used tests and stresses will be presented. Then, the test results for big arrays as well as for small arrays will be given. The section concludes with a discussion and a comparison of the results.

A. Used Tests and Stresses

The tests used during the experiment consist of interword base tests (i.e., test algorithms) and intraword base tests.

1) Interword Base Tests (Inter-BTs): These consist of the best-known industrial memory test algorithms; the tests are listed in Table VII. For Hammer, the notation, e.g., $10 * w1$ means that the write operation is performed ten times successively to the same cell. For GalRow and GalColumn, the notation, e.g., row$(r0, r1_b)$ means apply a $r0$ operation in an incrementing order to the cells of the row of the base cell, and apply $r1$ operation to the base cell after each $r0$ operation; a similar explanation applies to col$(r0, r1_b)$. Similarly, for WalkRow and WalkColumn, the notation, e.g., row$(r0)$ means apply a $r0$ operation using an incrementing address order to the row of the base cell, and skip the base cell; a similar explanation applies to col$(r0)$.

The inter-BTs have been applied using different stress combinations (SCs). Two types of stresses have been used, namely, addressing and DB stresses. The used addressing stresses consist of two types of addressing.

1. **Fast X (fx)**: Fast X addressing is simply incrementing or decrementing the address in such a way that each step goes to the next row.
2. **Fast Y (fy)**: Fast Y addressing is simply incrementing or decrementing the address in such a way that each step goes to the next column.

TABLE VIII
LIST OF USED INTRA-BTs

#	Test	# of operations
1	Test CFds	10+11$*\lfloor log_2 B \rfloor$
2	Test CFdr	6+6$*\lfloor log_2 B \rfloor$
3	Test CFtr	16* $\lfloor \log_2 B \rfloor$
4	Test CFwd	6+6$*\lfloor log_2 B \rfloor$
5	March SAM	28*$\lfloor log_2 B \rfloor$)
6	March SAMopt	28
7	Intra-MATS+	5+5*$\lfloor log_2 B \rfloor$

DB is defined as the pattern of ones and zeros as seen in an array of memory cells. The used DBs with the interword base tests are:

1. **solid** (s): All 0s, all 1s;
2. **checkerboard** (c): 0101 . . . /1010 . . . /0101 . . . /1010 . . .;
3. **column strip** (cs): 0101 . . . /0101 . . . /0101 . . . /01 010 . . .;
4. **row strip** (rs): 0000 . . . /1111 . . . /0000 . . . /1111 . . ..

2) Intraword Base Tests (Intra-BTs): These consist of the the new introduced tests in this paper, namely, Test CFds, Test CFdr, Test CFtr, Test CFwd, and March SAM and its optimized version (March SAMopt). In addition, and in order to have a reference for the comparison within intraword tests, intraword MATS+ (denoted Intra-MATS+) will be used; it is the same as MATS+ except that the test is repeated with $1 + \lfloor \log_2 B \rfloor$ different DBs; where B is the word size. For example for $B = 4$, the DBs will be 0000, 0101, and 1100. Note that the number of operations required for intraword MATS+ is $5 * (1 + \lfloor \log_2 B \rfloor)$ since MATS+ consists of five operations. Table VIII gives the list of the used intra-BTs together with the required number of operations. To find the test length of each BT, the required number of operations has to be multiplied with (n/B), where n is the size of the memory array. The intra-BTs of Table VIII, which use predefined DBs, has been applied using the two addressing types: fast x and fast y.

B. Test Results for Big Arrays

In the experiment done at Intel, a set of more than 70 tests was used. A test consists of a base test (BT) (i.e., test algorithm) applied using a particular stress combination (SC). A SC consists of a combination of values of different *stresses*; e.g., addressing, DBs, etc.

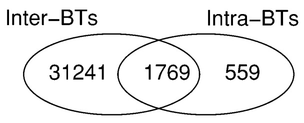

Fig. 9. FC for big arrays.

All tests were applied to embedded caches with a size of 1 MByte, which have an interleaved bit-organization; the testing has been performed at high voltage and high speed (about 15% more than the nominal values). From a huge number of tested chips, 344 chips failed all SCs, while 33 569 chips failed only some SCs. We will concentrate on the 33 569 chips since they are the most important.

Fig. 9 shows the Venn-diagram of the fault coverage (FC) of the inter-BTs as compared with the FC and intra-BTs. Apparently 33 010 out of 33 569 faults are detected with inter-BTs; 31 241 faults can be detected with inter-BTs *only*. On the other hand, 2328 faults are detected with intra-BTs; 559 faults are detected with intra-BTs *only*. Note that 1769 faults are detected with both inter-BTs and intra-BTs.

Based on the Venn-diagram one can conclude that the percentage of detected faults with inter-BTs is very high as compared with those detected with intra-BTs. Further, the percentage of faults detected with intra-BTs *only* (i.e., $559/33\,569 = 1.67\%$) cannot be ignored. Therefore, intraword faults have to be taken into consideration or leave a substantial number of faults undetected. Considering a high-volume production and the low DPM level driven by the market requirements, the percentage of such (intra-)word faults (e.g., 1.67% in our case) can translate into a high DPM level ending up in selling defective chips to the customers.

Analysis of Intra-BTs: This section compares the new intraword tests, including intra-MATS+. Our analysis will be focused only on faults detected with the Intra-BTs (i.e., the 559 faulty chips) (see Fig. 9).

Table IX shows the union and the intersections of the intra-BTs. A die belongs to the union of two BTs if at least one of the two BTs found the die to be faulty, and belongs to the intersection of two BTs if *both* BTs found the die to be faulty. The first column in the table gives the BT number; the second column the name of the BT. The column “SC” gives the addressing the BT is used with; not that each BT is applied with fx and fy addressing.[1] The column “FC” lists the fault coverage of the corresponding BT; the column “UFs” gives number of unique faults (UFs) each BT detects. Unique faults are faults that are only detected once with a single test.

The union and the intersection of each pair of BTs is shown in the rest of the tables. The numbers on the diagonal give the FC of the BTs, which are also listed in the column “FC;” e.g., March SAM with fx has FC = 312. The part above the main diagonal shows the union for each BT pair, while the part under the diagonal lists the intersection of each BT pair; for example, the union of Test CFdr with fx and March SAMopt with fx is 289 and their intersection is 144.

[1]Except fy for March SAM and intra-MATS+, these tests were unfortunately incorrectly implemented

Based on the test result database, Table IX and the Venn-diagram of Fig. 9, we conclude the following.

1) The total number of faulty chips detected with all intra-BTs is 2328 (see Fig. 9), i.e., 6.93% of the total 33 569 faulty chips.
2) The total number of faulty chips detected with intra-BTs *only* is 559, i.e., 1.67% of the total 33 569 faulty chips (see Fig. 9); 558 of such 559 faults are detected with the *new* introduced intra-BTs (Tests 1–6 in Table VI) as Fig. 10 shows.
3) The total number of faulty chips detected with the *new* introduced intra-BTs *only* is 347 (see Fig. 10), which is equivalent to:
a) 62.20% of the 559 faulty chips detected with intra-BTs *only* (see Fig. 10).
b) 14.90% of the of the total of 2328 faulty chips detected with *all* intra-BTs;
c) 1.03% of the total 33 569 faulty chips detected with all used tests (i.e., inter- and intra-BTs).
4) The best three Intra-BTs in terms of FC are: March SAM using fx with FC = 312, and Test CFds using fx with FC = 281 and Test CFds using fy with FC = 271; this is exactly what we expected based on our theoretical analysis (see Section IV-C).
5) The best union pair in terms of the FC is 383, which is achieved with intra-MATS+ using fx and March SAM using fx. However, all detected faults by intra-MATS+ are detected with the new introduced intra-BTs except one fault as Fig. 10 shows.
6) There are four UFs detected with four tests (see column “UFs” in Table IX).
7) An analysis (not shown here) reveals that in order to achieve the same FC as that achieved with all used intra-BTs, only the following intra-BTs are required: Test CFds-fx, Test CFwd-fx, March SAM-fx, and intra-MATS+ using fx. These tests are the tests detecting UFs (see Table IX).
8) A test with the lowest FC is Test CFwd with FC = 132 for fx and FC = 127 for fy.
9) None of the intra-BTs detects supersets of faults of other intra-BTs.

C. Test Results for Small Arrays

A similar experiment has been done for small arrays with a size of 25 KB, and with an *adjacent* bit-organization. The same tests at the same test conditions (i.e., high voltage and high speed) as for big arrays have been applied. From a huge number of tested chips (which is *1.5 times more* than the number of big array chips tested in the previous experiment), 2490 chips failed all SCs, while 6039 chips failed only some SCs. We will concentrate on the 6039 chips since they are the most important.

Fig. 11 shows the Venn-diagram of the FC of the inter-BTs as compared with the FC and intra-BTs. Apparently, 5982 out of 6039 faults are detected with inter-BTs; 475 faults can be detected with inter-BTs *only*. On the other hand, 5564 faults are detected with intra-BTs; 57 faults are detected with intra-BTs

TABLE IX
UNION AND THE INTERSECTION OF INTRA-BTS FOR BIG ARRAYS (TOTAL FC $= 559$)

#	BT Name	SC	FC	UFs	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Test CFdr	fx	239	0	**239**	288	347	337	307	315	270	266	289	305	376	–	347	–
2	Test CFdr	fy	217	0	168	**217**	324	314	291	296	246	245	264	275	355	–	324	–
3	Test CFds	fx	281	0	173	174	**281**	339	344	355	305	303	313	330	375	–	364	–
4	Test CFds	fy	271	1	173	174	213	**271**	334	343	287	290	305	313	362	–	349	–
5	Test CFtr	fx	218	0	150	144	155	155	**218**	282	235	240	281	300	361	–	313	–
6	Test CFtr	fy	231	0	155	152	157	159	167	**231**	251	244	292	307	368	–	319	–
7	Test CFwd	fx	132	1	101	103	108	116	115	112	**132**	155	230	251	336	–	250	–
8	Test CFwd	fy	127	0	100	99	105	108	105	114	104	**127**	226	247	329	–	247	–
9	March SAMopt	fx	194	0	144	147	162	160	131	133	96	95	**194**	257	338	–	297	–
10	March SAMopt	fy	215	0	149	157	166	173	133	139	96	95	152	**215**	345	–	314	–
11	March SAM	fx	312	1	175	174	218	221	169	175	108	110	168	182	**312**	–	383	–
12	March SAM	fy	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
13	Intra-MATS+	fx	212	1	104	105	129	134	117	124	94	92	109	113	141	–	**212**	–
14	Intra-MATS+	fy	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

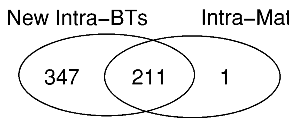

Fig. 10. FC of intra-BTs for big arrays.

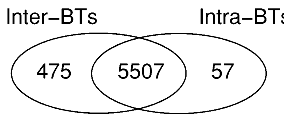

Fig. 11. FC for small arrays.

only. Note that 5507 faults are detected with both inter-BTs and intra-BTs.

Based on the Venn-diagram one can conclude that the percentage of the faults detected with intra-BTs *only* is very small (about 0.94%) as compared with those detected with inter-BTs (about 99.06%).

Analysis of Intra-BTs: Table X shows the *union* and *intersections* of the intra-BTs. The table focuses only on faults detected with the intraword tests (i.e., the 57 faulty chips) (see Fig. 11). The representation used in Table X is similar to that used in Table IX. Based on the test result database (Table X and the Venn-diagram of Fig. 11), we conclude the following.

1) The total number of faulty chips detected with all intra-BTs is 5564 (see Fig. 11), i.e., 92.13% of the total 6039 faulty chips.
2) The total number of faulty chips detected with intra-BTs *only* is 57, i.e., 0.94% of the total 6039 faulty chips (see Fig. 11). 53 of such 57 faults are detected with the *new* introduced intra-BTs (Test 1 through Test 5 in Table VI) as Fig. 12 shows.
3) The total number of faulty chips detected with the *new* introduced intra-BTs *only* is 4 (see Fig. 12), which is equivalent to:
   a) 7.01% of the 57 faulty chips detected with intra-BTs *only* (see Fig. 12).
   b) 0.95% of the of the total of 5564 faulty chips detected with *all* intra-BTs.
   c) 0.88% of the total 6039 faulty chips detected with all used tests (i.e., inter- and intra-BTs).
4) The best intraword tests in terms of FC are: intra-MATS+ using fx with FC $=$ 52 and Test CFtr using x with FC $= 52$. However, the FC of all intra-BTs varies between 50 and 52. Note that the variation in the FC for small arrays is very small as compared with big arrays. For the big arrays, the FC of intra-BTs varies between 127 and 312 as shown in Table IX.
5) The best union pair in terms of the FC is 55, which is achieved with intra-MATS+ using fx and one of the following tests: Test CFdr (using fx or fy), CFtr using fx, or CFwd using fx.
6) There are five UFs detected with two tests: Test CFtr-fx with UFs $= 1$ and intra-MATS+-fx with UFs $= 4$ (see column "UFs" in Table X).
7) By inspecting Table X, we can see that the intra-BTs can be divided into four groups with each group consists of BTs that have *exactly the same* FC. The four groups are given in the following table.

Groups	Tests
G1	CFdr-fx. CFdr-fy. CFwd-fx
G2	CFds-fx. CFds-fy. CFtr-fy. SAMopt-fx
G3	CFwd-fy. SAMopt-fy. SAM-fx
G4	CFtr-fx
G5	intra-MATS+-fx

Table X can now be presented in a compact form without loss of information. The compact presentation is given in Table XI; a similar representation is used. From Table XI one can clearly see that G4 detect all faults detected with G2 and G3. Therefore, using one test from G1, the test of G4 and the test of G5 will achieve the same FC as that achieved by the initial used intra-BTs; i.e., in order to achieve the same FC as that achieved with all intra-BTs, only the following three tests are required: intra-MATS+-fx, Test CFtr-fx, and, e.g., Test CFdr-fx.

TABLE X
UNION AND THE INTERSECTION OF INTRA-BTs FOR SMALL ARRAYS (TOTAL FC = 57)

#	BT Name	SC	FC	UFs	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Test CFdr	fx	51	0	**51**	51	52	52	53	52	51	51	52	51	51	–	56	–
2	Test CFdr	fy	51	0	51	**51**	52	52	53	52	51	51	52	51	51	–	56	–
3	Test CFds	fx	51	0	50	50	**51**	51	52	51	52	51	51	51	51	–	55	–
4	Test CFds	fy	51	0	50	50	51	**51**	52	51	52	51	51	51	51	–	55	–
5	Test CFtr	fx	52	1	50	50	51	51	**52**	52	53	52	52	52	52	–	56	–
6	Test CFtr	fy	51	0	50	50	51	51	51	**51**	52	51	51	51	51	–	55	–
7	Test CFwd	fx	51	0	51	51	50	50	50	50	**51**	51	52	51	51	–	56	–
8	March CFwd	fy	50	0	50	50	50	50	50	50	50	**50**	51	50	50	–	55	–
9	March SAMopt	fx	51	0	50	50	51	51	51	51	50	50	**51**	51	51	–	55	–
10	March SAMopt	fy	50	0	50	50	50	50	50	50	50	50	50	**50**	50	–	55	–
11	March SAM	fx	50	0	50	50	50	50	50	50	50	50	50	50	**50**	–	55	–
12	March SAM	fy	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
13	Intra-MATS+	fx	52	4	47	47	48	48	48	48	47	47	48	47	47	–	**52**	–
14	Intra-MATS+	fy	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

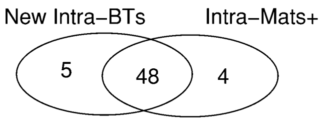

Fig. 12. FC of intra-BTs for small arrays.

TABLE XI
COMPACT REPRESENTATION OF TABLE X

Group	FC	UFs	G1	G2	G3	G4	G5
G1	51	0	**51**	52	51	53	56
G2	51	0	50	**51**	51	52	55
G3	50	0	50	50	**50**	52	55
G4	52	1	50	51	50	**52**	56
G5	52	4	47	48	47	48	**52**

TABLE XII
SUMMARY OF THE TEST RESULTS

	Big Arrays	Small Arrays
Size	1MB	25KB
# of chips tested	*N*	1.5**N*
# chips failed	33913	8529
# chips failing *all* tests	344	2490
# chips failing *some* tests	33569	6039

TABLE XIII
COMPARISON OF THE TEST RESULTS

Faulty chips detected with	Big Arrays		Small arrays	
	#	%	#	%
Inter and Intra BTs	33569	100.00	6039	100.00
Inter-BTs	33010	98.33	5982	99.06
Intra-BTs	2328	6.93	5564	92.21
Intra-BTs *only*	559	1.67	57	0.94
The new Intra-BTs	558	1.66	53	0.88
The new Intra-BTs *only*	347	1.03	5	0.08

D. Comparison and Analysis of the Test Results

From a theoretical point of view, we expect that intraword CFs for small memory which have an adjacent bit-organization will be more important than for big arrays with interleaved bit-organization (see Section IV-D). However, our (limited) experiment results show the contrary result as will be described below.

Table XII summarizes the test results for the big and the small arrays used in our experiment. Remember that the same tests under the same test conditions have been used for both types of arrays. As the table shows, the number of small array chips tested is 1.5 times more than that of big arrays. However, the number of detected faults for big array is about four times more than that of small arrays. This indicates clearly that as the size of the memory increases, the sensitivity to the faults also increases. In addition, the percentage of the detected faults with all used tests (i.e., the intersection of all used tests) is about 0.01% of the faults for big array, while this is about 29.92% for small arrays. The common faults that can be detected with all tests are the traditional well-known (easy-to-detect) faults like SAF, TFs, etc. These faults are possibly the dominant ones for small arrays. However, for big arrays, the faults manifest themselves in more complicated and different ways; this is indicated by the percentage of the common faults detected with all used tests (i.e., 0.01%).

Table XIII compares the numbers of faults detected with inter- and intra-BTs for the two arrays; only the faults that do not fail *all* tests are considered. The percentage of the detected faults with intra-BTs *only* (i.e., unique faults for intra-BTs) is about 1.67% of the total faults for big arrays and 0.94% for small arrays. If we assume that the (main) targeted faults by intra-BTs are intraword CFs, then we can conclude that occurrence probability of intraword CFs for big arrays (interleaved bit-organization) is much higher than for small arrays (adjacent bit-organization); this conclusion is contrary with what we would have expected from theoretical point of view. The explanation of this conclusion is given next.

The intraword CFs can occur between cells belonging to the same word (adjacent) as well as between I/O data paths. Such paths are the signals with high fanout like word lines, bit lines, and address decoder preselect lines. The lines carrying those signals run across the memory area and therefore have, in addition to high load, also a high-capacitance coupling and crosstalk effect with other signal, power, and ground lines. It is evident that a big array has long lines and therefore a higher capacitive coupling between those lines than for small array. This means that the big arrays are more sensitive for intraword CFs between theses lines than for small arrays. Based on our experiment, we can conclude that intraword CFs between I/O paths are more important than between adjacent memory cell within a word.

VI. Conclusion

In this paper, all possible intraword CFs have been presented. The required DBS and the operation sequences (OSs) to detect each of the CFs have been presented. This resulted into four intraword base tests. A DBS and OS to detect all intraword CFs have been also presented and compiled into a single test detecting all faults. The latter test has also been optimized in terms of test time when the intraword CFs are restricted to physically adjacent cells within a word.

Thereafter, the result of an industrial evaluation of the presented base tests for intraword CFs have been presented and compared with 15 known memory tests. The set of base tests have been applied to small memories (25 KB) with bit-adjacent organization as well as to big memories (1 MB) with bit-interleaved organization. The following conclusions can be drawn for the memories considered in the experiment.

1) The sensitivity of the memory to the faults increases with the increase in its size. In our experiment, the number of detected faults for a big array is about four times more than those for small array. The big array is about 40 times bigger than the small one, and the number of tested small arrays chips was 1.5 times more than that of tested big arrays.
2) That the adjacent memory arrays are more sensitive for intraword CFs than interleaved memory arrays is an incorrect statement. Our experiment shows the contrary. This is possibly due to the fact that the intraword CFs does not only occur due to the CFs between cells of a single word, but also due to the coupling between adjacent lines running across the memory array like the bit line and the word lines.
3) Intraword CFs should be considered for any serious test purpose or leave substantial defects (significant DPM) undetected. The percentage of detected faults with intra-BTs only (designed to target intraword faults), which is about 1.67% for big arrays, cannot be ignored, especially when considering high volume production and a very low DPM level.

The above results are, of course, design-dependent. For other memory designs and/or implementations, different results may be expected. Also, because the size of the experiment was limited, some effects may not have been noticed.

References

[1] R. D. Adams, *High Performance Memory Testing*. Norwell, MA: Kluwer, 2003.
[2] M. S. Abadir and J. K. Reghbati, "Functional testing of semiconductor random access memories," in *ACM Comput. Surveys*, vol. 15, 1983, pp. 175–198.
[3] Z. Al-Ars and Ad. J. van de Goor, "Static and dynamic behavior of memory cell array opens and shorts in embedded DRAMs," in *Proc. Design Automation Test Eur.*, 2001, pp. 496–503.
[4] M. A. Breuer and A. D. Friedman, *Diagnosis and Reliable Design of Digital Systems*. Woodland Hills, CA: Comput. Sci. Press, 1976.
[5] R. Dekker *et al.*, "A realistic fault model and test algorithms for static random access memories," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 9, no. Jun., pp. 567–572, 1990.
[6] J. H. De Jonge and A. J. Smeulders, "Moving inversions test pattern is thorough, yet speedy," *Comput. Design*, pp. 169–173, 1976.
[7] S. Hamdioui and A. J. van de Goor, "Experimental analysis of spot defects in SRAMs: Realistic fault models and tests," in *Proc. 9th Asian Test Symp.*, 2000, pp. 131–138.
[8] S. Hamdioui, Z. Al-Ars, and A. J. van de Goor, "Testing static and dynamic faults in random access memories," in *Proc. IEEE VLSI Test Symp.*, 2002, pp. 395–400.
[9] S. Hamdioui, A. J. van de Goor, and M. Rodgers, "March SS: A test for all static simple RAM faults," in *Proc. IEEE Int. Workshop Memory Technol., Design, Testing*, 2002, pp. 95–100.
[10] S. Hamdioui, *Testing Static Random Access Memories: Defects, Fault Models, and Test Patters*. Norwell, MA: Kluwer, 2004.
[11] M. Marinescu, "Simple and efficient algorithms for functional RAM testing," in *Proc. Int. Test Conf.*, 1982, pp. 236–239.
[12] R. Nair, "An optimal algorithm for testing stuck-at faults random access memories," *IEEE Trans. Comput.*, vol. C-28, no. 3, pp. 258–261, Mar. 1979.
[13] I. Schanstra and A. J. van de Goor, "Industrial evaluation of stress combinations for march tests applied to SRAMs," in *Proc. IEEE Int. Test Conf.* , 1999, pp. 983–992.
[14] D. S. Suk and S. M. Reddy, "A march test for functional faults in semiconductor random-access memories," *IEEE Trans. Comput.*, vol. C-30, no. 12, pp. 982–985, Dec. 1981.
[15] P. R. Treurer and V. K. Agarwal, "Fault location algorithms for repairable embedded RAMs," in *Proc. IEEE Int. Test Conf.*, 1993, pp. 825–834.
[16] A. J. van de Goor, M. S. Abadir, and A. Carlin, "Minimal test for detecting state coupling faults memories," in *Proc. Design Automation Test Eur.*, 2002, pp. 944–948.
[17] 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. IEEE Int. Workshop Memory Technol., Design, Testing*, 1998, pp. 46–52.
[18] A. J. van de Goor and A. Paalvast, "Industrial evaluation of DRAM's SIMM tests," in *Proc. Int. Test Conf.*, 2000, pp. 426–435.
[19] A. J. van de Goor and I. B. S. Tlili, "March tests for word-oriented memories," in *Proc. Design Automation Test Eur.*, 1998, pp. 501–508.
[20] A. J. van de Goor and Z. Al-Ars, "Functional fault models: A formal notation and taxonomy," *Proc. IEEE VLSI Test Symp.*, pp. 281–289, 2000.
[21] A. J. van de Goor, *Testing Semiconductor Memories, Theory and Practice*. Gouda, The Netherlands: ComTex, 1998.

Said Hamdioui (S'99–M'01) received the M.S.E.E. and Ph.D. degrees (both with honors) from the Delft University of Technology, Delft, The Netherlands, in 1997 and 2001, respectively.

He is currently with the Delft University of Technology. He interned with Intel Corporation for over two years, while pursuing and after receiving the Ph.D. degree, and was responsible for developing new low-cost and efficient test algorithms for advanced Intel singe-port and multiport cache designs. He has published one book and more than 30 papers in the area of memory testing. His research interests concern systematic fault modeling, test generation and optimization for semiconductor memories, yield enhancement, product engineering, etc.

Dr. Hamdioui was the recipient of the European Design Automation Association (EDAA) Award for 2003.

John Eleazar Q. Delos Reyes received the M.S. degree in electronics and communications engineering from Saint Louis University, Baguio City, Philippines, in 1996.

He was with Intel Philippines, where he worked on cache components as a Product Engineer for Pentium II microprocessors. Since then, he continues to be a Product Engineer focusing on embedded cache testing of microprocessors with the Intel Corporation, Hillsboro, OR.