Optimizing Test Length for Soft Faults in DRAM Devices

Zaid Al-Ars Said Hamdioui Georgi Gaydadjiev

Delft University of Technology, Faculty of EE, Mathematics and CS

Laboratory of Computer Engineering, Mekelweg 4, 2628 CD Delft, The Netherlands

E-mail: z.al-ars@tudelft.nl

Abstract: *Soft faults in DRAMs are faults that do not get sensitized directly after an operation is performed, but require a time to pass before the fault can be detected. Tests developed to detect these faults are rather complex and take an exceptionally long time to apply on the memory. This paper discusses a number of methods to optimize the test length for soft faults, based on the electrical design of the memory and the topology of the layout. These methods make it possible to reduce the delay time needed in the test such that it does not scale with the number of cells in the memory.*

Keywords: DRAM testing, soft faults, test length optimization, memory layout, circuit design, delay time.

1 Introduction

DRAMs exhibit special time dependent faulty behavior that may delay the sensitization of a fault by a given period of time, before the fault can become detectable by a read operation. Faults of this type are referred to as *soft faults*, which take place due to naturally occurring leakage mechanisms in the storage cell that deplete the stored data within the cell causing it to fail after some time [Al-Ars04]. Tests proposed to detect soft faults are very time consuming, and overly expensive to implement in practice [Al-Ars06]. The time complexity of these tests scale with a factor of $n \cdot T$, where n is the number of cells in the memory and T is the retention time of the memory (typically $T > 64$ ms). This complexity limits the practicality of these tests to the industry.

This paper discusses a number of methods to optimize the test length of soft fault tests to make them more applicable in practice. Design based optimizations can reduce the length of single-cell tests from linear to constant with respect to the retention time in the memory. The same is true for two-cell tests, where the topological location of the cells with respect to each other on the layout is used to reduce the test length.

This paper is organized as follows. Section 2 presents memory fault models in general, followed by relevant DRAM-specific faults in Section 3. Section 4 derives the tests needed to detect soft faults in DRAMs and identifies their complexity. Then, Section 5 discusses the methods used to reduce the test length and derives the new optimized tests. Finally, Section 6 ends with the conclusions.

2 Fault primitives

In order to specify a certain memory fault, one has to represent it in the form of a *fault primitive (FP)*, denoted as $<S/F/R>$. S describes the operation sequence that sensitizes the fault, F describes the logic level in the faulty cell ($F \in \{0, 1\}$), and R describes the logic output level of a read operation ($R \in \{0, 1, -\}$). R has a value of 0 or 1 when the fault is sensitized by a read operation, while the "$-$" is used when a write operation sensitizes the fault. For example, in the FP $<0w1/0/->$, which is the up-transition fault (TF_1), $S = 0w1$ means that a $w1$ operation is written to a cell initialized to 0. The fault effect $F = 0$ indicates that after performing $w1$, the cell remains in state 0. The output of the read operation ($R = -$) indicates there is no expected output for the memory.

Functional fault models (FFMs) can be defined as a non-empty set of FPs. The most important FFM classes are single-cell static FFMs and two-cell static FFMs.

Single-cell static FFMs consist of FPs sensitized by performing at most one operation on a faulty cell. Table 1 lists all single-cell static FFMs and their corresponding FPs. In total, there are 6 different types of FFMs: state fault (SF), transition fault (TF), write destructive fault (WDF), read destructive fault (RDF), incorrect read fault (IRF), deceptive read destructive fault (DRDF) [Adams96].

Table 1. Single-cell static FFMs and their corresponding FPs.

#	Fault	FP	Name
1	SF	$<0/1/->$, $<1/0/->$	State fault
2	TF	$<0w1/0/->$, $<1w0/1/->$	Transition fault
3	WDF	$<0w0/1/->$, $<1w1/0/->$	Write destructive fault
4	RDF	$<0r0/1/1>$, $<1r1/0/0>$	Read destructive fault
5	IRF	$<0r0/0/1>$, $<1r1/1/0>$	Incorrect read fault
6	DRDF	$<0r0/1/0>$, $<1r1/0/1>$	Deceptive RDF

Two-cell static FFMs consist of FPs sensitized by performing at most one operation while considering the faulty effect of two cells. Such FPs can be represented as $<S_a; S_v/F/R>$, where S_a is the sequence performed on the aggressor (a) and S_v is the sequence performed on the victim (v). Table 2 lists all two-cell static FFMs and their corresponding FPs. In total, there are 7 different types of two-cell static FFMs.

25th IEEE VLSI Test Symmposium (VTS'07)
0-7695-2812-0/07 $20.00

Table 2. Two-cell static FFMs and their FPs ($x, y \in \{0, 1\}$).

#	Fault	FP	Name
1	CFst	$<0;0/1/->$, $<0;1/0/->$	State coupling
		$<1;1/0/->$, $<1;0/1/->$	fault
2	CFds	$<xwy;0/1/->$, $<xwy;1/0/->$	Disturb coupling
		$<xrx;0/1/->$, $<xrx;1/0/->$	fault
3	CFtr	$<0;0w1/0/->$, $<0;1w0/1/->$	Transition coupling
		$<1;0w1/0/->$, $<1;1w0/1/->$	fault
4	CFwd	$<0;0w0/1/->$, $<0;1w1/0/->$	Write destructive
		$<1;0w0/1/->$, $<1;1w1/0/->$	coupling fault
5	CFrd	$<0;0r0/1/1>$, $<0;1r1/0/0>$	Read destructive
		$<1;0r0/1/1>$, $<1;1r1/0/0>$	coupling fault
6	CFir	$<0;0r0/0/1>$, $<0;1r1/1/0>$	Incorrect read
		$<1;0r0/0/1>$, $<1;1r1/1/0>$	coupling fault
7	CFdrd	$<0;0r0/1/0>$, $<0;1r1/0/1>$	Deceptive read
		$<1;0r0/1/0>$, $<1;1r1/0/1>$	destructive CF

3 DRAM-specific faults

DRAM faults can either be attributed to leakage currents (resulting in time dependent faults), or to improperly set voltages (resulting in voltage dependent faults). Figure 1 shows a summary of DRAM-specific faults.

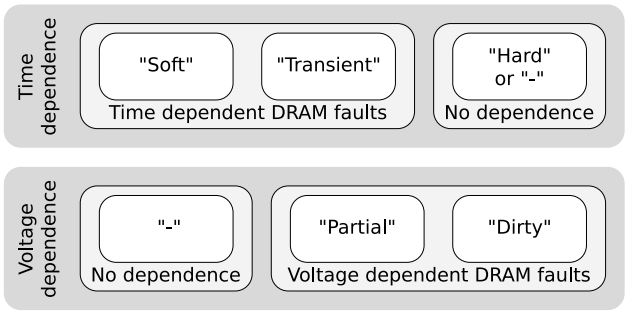
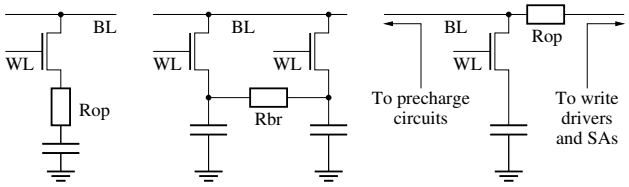


Figure 1. Summary of the space of DRAM-specific faults.

3.1 Time dependent faults

Time dependent faults are caused by leakage currents in faulty cells [Keshavarzi97]. Time dependence divides all faults into three classes: soft, transient and hard.

Soft faults—Soft faults (s) only become detectable after some time from their sensitization. These faults can be tested for by adding a *delay* within the test, as it is the case for the *data retention fault*, for example [Dekker90]. Soft faults are caused by writing weak voltages into memory cells, that soon get depleted by naturally occurring leakage currents. Soft faults are represented as sFP $=<S_T/F/R>$, where S has an added time parameter T to indicate that some time should elapse before full sensitization. The open defect in Figure 2(a) shows an open that may cause soft faults in a DRAM cell. If the open defect has an intermediate resistance value that is not too high (causing hard faults) and not too low (not causing a fault at all), write operations store a *weak voltage* into the cell. If leakage opposes the weak voltage, the stored information gets lost over time.



(a) Partial in I (b) Partial in A (c) Dirty faults

Figure 2. Defects causing (a) p_i, (b) p_a, and (c) causing dirty faults.

Transient faults—Transient faults (t) are memory faults that do not remain sensitized indefinitely, but tend to correct themselves after a period of time. We will not discuss transient faults in this paper.

Hard faults—Identifying a fault as being hard ("h" or "-") indicates that it is neither soft nor transient (i.e., it is insensitive to time). All the generic faults described in Section 2 are hard faults.

3.2 Voltage dependent faults

Operations performed on a defective DRAM may set improper voltage levels on memory nodes, thereby causing two types of DRAM faults: partial faults and dirty faults.

Partial faults—Partial faults (p) are faults that can only be sensitized when a specific memory operation is successively repeated a number of times, either to properly initialize the faulty cell (*partial faults during initialization* p_i), or to properly sensitize the fault in the cell (*partial faults during fault sensitization or activation* p_a). Figure 2(a) shows an example of an open (R_{op}) in the cell, causing p_i. R_{op} prevents fully initializing the cell to the required voltage with only one operation, which means that full initialization requires repeating the operation a number of times. Figure 2(b) shows an example of a bridge (R_{br}) between two cells, causing p_a. These faults are modeled by performing an operation Ox an h (or *hammer*) number of times. For example, if $<xOy/F/R>$ becomes partial during initialization p_i, it should be modeled as p_iFP $=<x^hOy/F/R>$.

Dirty faults—Dirty faults (d) assume that after proper initialization or sensitization, the state of the memory (voltages on the BLs, the WLs, or in data buffers) is corrupted, such that subsequent detection is prevented. In order to ensure detectability, additional operations (so called *completing operations*) must be performed to correct the corrupted state of the memory. Figure 2(c) shows an example of an open defect (R_{op}) on the BL that causes dirty faults. This defect disconnects memory cells from the write drivers, which prevents the memory from writing the cells. This

defect also prevents properly precharging the BL. As a result, a $w0$ operation that fails to write 0 in the cell ends up preconditioning the BL to properly sense a 0, thereby causing a dirty fault. These fault are modeled by the introduction of completing operations to the FP. Detectability of all known dirty faults can be ensured using a completing write operation with data opposite to the data in the victim, performed to a cell different from the victim but positioned on the same BL pair (i.e., dFP $=<xO_vy[w_b\overline{y}]/\overline{y}/->_{b,v\in \mathrm{BL}}$).

3.3 Realistic space of DRAM faults

Any generic memory fault, described in Section 2, can represent a DRAM-specific fault by adding DRAM-specific fault attributes to it. First, there are voltage dependent attributes: partial (p), dirty (d), or neither (-). Second, there are time dependent attributes: hard (h or -), soft (s) and transient (t). Furthermore, the partial attribute can either be initialization related (p_i), or activation (or sensitization) related (p_a).

Based on a detailed analysis of the characteristics of these faults, the full realistic space of DRAM faults can be constructed for singe-cell faults, as well as two-cell faults [Al-Ars05].

$$\text{Single-cell fault} = \left\{ \begin{array}{c} \text{-} \\ p_i \\ d \\ p_id \end{array} \right\} \left\{ \begin{array}{c} \text{h or -} \\ s \\ t \end{array} \right\} \text{FP} \qquad (1)$$

$$\text{Two-cell fault} = \left\{ \begin{array}{c} \text{-} \\ p \end{array} \right\} \left\{ \begin{array}{c} \text{h or -} \\ s \\ t \end{array} \right\} \text{FP} \qquad (2)$$

These expressions indicate that any generic single-cell fault can either be regular (-), initialization partial (p_i), dirty (d) or partial dirty (p_id), while being hard (h or -), soft (s) or transient (t) at the same time. Two-cell faults can regular (-) or partial (p), while being hard, soft or transient. Note that some faults classes are considered unrealistic, such as activation partial (p_a) single-cell faults, and therefore they are not included in the space.

For example, a transition 0 fault can be hard (hTF_0), which is the same as the generic TF_0. It can also be partial hard (p_ihTF_0), dirty hard ($dhTF_0$) and partial dirty hard fault (p_idhTF_0) The same combinations apply for soft TF_0 and transient TF_0.

4 Detecting soft faults

Soft FPs mean that a correct (but weak) voltage in the cell can gradually be depleted and cause a detectable fault in the cell after a period of time.

Detection conditions for soft faults

An FP has two components to describe a fault: F and R. Only F can cause soft faults (fail after some time), whereas R cannot be soft, since its value is read at moment a read operation is performed.

Table 3. List of single-cell, soft FPs and their detection conditions. Ox_b is performed with a value (x) opposite to that in the sensitizing operation and to a cell (b) different from v, but along the same BL as v.

# Fault	FP (b belongs to BL of v)	Detection cond., $O \in \{w, r\}$
1 ds SF_0	$<0_v[O1_b]_T/1/->$	$\Updownarrow(..w0, ..O1_b, ..T, ..r0..)$
2 ds SF_1	$<1_v[O0_b]_T/0/->$	$\Updownarrow(..w1, ..O0_b, ..T, ..r1..)$
3 $p_i$ds WDF_0	$<w0^h_v[O1_b]_T/1/->$	$\Updownarrow(..w0^h, ..O1_b, ..T, ..r0..)$
4 $p_i$ds WDF_1	$<w1^h_v[O0_b]_T/0/->$	$\Updownarrow(..w1^h, ..O0_b, ..T, ..r1..)$
5 $p_i$ds TF_1	$<w0^h_v w1_v[O0_b]_T/0/->$	$\Updownarrow(..w0^h, ..w1, ..O0_b, ..T, ..r1..)$
6 $p_i$ds TF_0	$<w1^h_v w0_v[O1_b]_T/1/->$	$\Updownarrow(..w1^h, ..w0, ..O1_b, ..T, ..r0..)$
7 $p_i$ds IRF_0	$<w0^h_v[O1_b]r0_v/0/1>$	$\Updownarrow(..w0^h, ..O1_b, ..r0..)$
8 $p_i$ds IRF_1	$<w1^h_v[O0_b]r1_v/1/0>$	$\Updownarrow(..w1^h, ..O0_b, ..r1..)$
9 $p_i$ds $DRDF_0$	$<w0^h_v r0_v[O1_b]_T/1/0>$	$\Updownarrow(..w0^h, ..r0, ..O1_b, ..T, ..r0..)$
10 $p_i$ds $DRDF_1$	$<w1^h_v r1_v[O0_b]_T/0/1>$	$\Updownarrow(..w1^h, ..r1, ..O0_b, ..T, ..r1..)$
11 $p_i$ds RDF_0	$<w0^h_v[O1_b]r0_v/1/1>$	$\Updownarrow(..w0^h, ..O1_b, ..r0..)$
12 $p_i$ds RDF_1	$<w1^h_v[O0_b]r1_v/0/0>$	$\Updownarrow(..w1^h, ..O0_b, ..r1..)$

Table 3 lists all single-cell soft faults, along with the detection conditions needed to detect them. Note that the detection conditions for soft IRF and RDF do not include the T, since these faults are detected as soon as they get sensitized. As an example, the (partial, dirty and soft) transition 0 fault ($p_i$ds TF_0), must first be initialized a multiple number of times ($w1^h$). Then, the sensitizing write 0 operation can be performed ($w0$), before a completing operation with data 1 is applied to a different cell along the same BL ($[O1_b]$). To ensure the detection of this soft fault, a delay time T must be introduced after the completing operation to allow for sensitization to take place.

In the same way, one can derive the detection conditions corresponding to all two-cell, soft faults.

Tests for soft faults

Based on the detection conditions of single and two-cell soft faults, it is possible to derive memory tests that detect all these faults. A march test that detects all single-cell soft faults can have the form of March S1C below (for soft, single-cell).

March S1C = {
$\Updownarrow(w0^h, r0, w1_b, T, r0)$; $\Updownarrow(w1^h, r1, w0_b, T, r1)$;
ME0 ME1
$\Updownarrow(w0^h, w1, w0_b, T, r1)$; $\Updownarrow(w1^h, w0, w1_b, T, r0)\}$
ME2 ME3

This march test has 4 march elements (ME0 through ME3). This test is similar to the test for hard single-cell DRAM faults (March H1C), which is expected since the space of soft faults is derived from the space of hard faults. The test uses special nonstandard march elements, where and operation should be performed on a cell b different from the current cell the march element is accessing. This ensures that the BL has the opposite state as compared to the sensitized cell. The test has a complexity of $(12 \cdot n + 4 \cdot h \cdot n + 4 \cdot T \cdot n)$. For a typical retention time of $T > 64$ ms, the total idle test time is impractical. It is important to implement *design-for-testability* (*DFT*) techniques, or use test stresses that force soft faults to become directly detectable hard faults, which in turn do not require any delay time to detect [Al-Ars05, Wang01].

A march test that detects all two-cell soft faults can be represented by March S2C below.

March S2C = {
$\Updownarrow(w0^h)$; $\Uparrow(r0^h, T, r0, w1^h)$;
ME0 ME1
$\Uparrow(r1^h, T, r1, w0^h)$; $\Downarrow(r0^h, T, r0, w1^h)$;
ME2 ME3
$\Downarrow(r1^h, T, r1, w0^h)$; T; $\Updownarrow(r0)\}$
ME4 ME5 ME6

This march test has 7 march elements (ME0 through ME6). Note that ME5 is simply a single delay T added to sensitize the faults before the final detecting read operations are performed in ME6. This test has a time complexity of $[5 \cdot n + 9 \cdot h \cdot n + (4 \cdot n + 1)T]$, which is prohibitively expensive due to the $(4 \cdot n + 1)T$ term. A number of test time reduction methods need to be implemented, such as test stresses, which force soft faults to become directly detectable hard faults [Al-Ars05, Wang01].

5 Test length reduction

The tests derived in Section 4 assume the worst case faulty behavior possible in the memory. In such a situation, the test engineer has no knowledge of the internal structure, the design or the layout of the memory under test. Therefore, all possible faults should be tested for in order to ensure the best fault coverage. When the internal structure is known, this information can be used to reduce the complexity of the used tests. In this section, we first discuss using electrical design information, then layout information to reduce the tests.

5.1 Memory design consideration

The example of dirty faults discussed in Section 3 above shows that, in order for dirty faults to take place, the precharge circuitry should be located on one side of the BL pair, while the sense amplifiers located on the other side [see Figure 2(c)]. The memory can, however, be designed in two different ways.

Table 4. Different positions of the SA and PR on either side of a BL pair.

#	Upside		Downside		Description
	SA	PR	SA	PR	
1	t	t	b	b	SA and PR on same side of BL
2	t	b	b	t	SA on one side, PR on other

Table 4 lists the 4 different combinations of positions for the sense amplifiers (SA) and the precharge circuit (PR) on either side of a BL pair. An entry "t" means that the corresponding circuitry is located at the top of the BL pair, while an entry "b" means that the corresponding circuitry is located at the bottom of the BL pair. Since the top and the bottom of a BL pair are symmetrical, only 2 combinations in Table 4 are unique, while the other 2 combinations can be derived by viewing the BL pair from the upside or from the downside.

For example, Figure 3 shows one possible position allocation of the SA and PR circuitry. In this configuration, the SA is located on one side of the BL pair and the PR is located on the other side. Considering the upside of the BL pair, the SA is located at the top while the PR is located at the bottom. Considering the downside of the BL pair, the PR is located at the top while the SA is located at the bottom. Obviously, both of these BL organizations are identical.

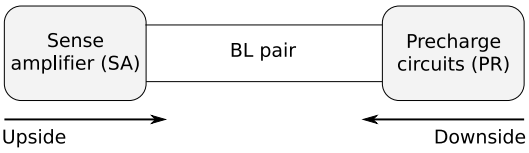


Figure 3. Upside and downside symmetry of BL pairs.

Dirty faults only take place when SA and PR are located on different ends of the BL, which is combination #2 in Table 4. However, most DRAMs today are designed with SA and PR located on the same side of the BL (combination #1 in Table 4), in which case dirty faults are not possible [Al-Ars05]. In other words, this organization reduces the space of single-cell memory faults exhibited as a result of all possible defects from the one described in Expression 1 to the following one.

$$\text{Single-cell fault} = \left\{ \begin{matrix} - \\ p_i \end{matrix} \right\} \left\{ \begin{matrix} h \\ s \\ t \end{matrix} \right\} \text{FP} \qquad (3)$$

The reduction in the space of single-cell faults implies a corresponding simplification in the tests for these faults,

since dirty faults need not be tested for. March S1C$_{part}$ ("*part*" for "only partial") below detects all single-cell soft faults for this special case.

March S1C$_{part}$ = {

$\Updownarrow(w0^h, r0);$	$T;$	$\Updownarrow(r0);$	$\Updownarrow(w1^h, r1);$	$T;$	$\Updownarrow(r1);$
ME0	ME1	ME2	ME3	ME4	ME5
$\Updownarrow(w0^h, w1);$	$T;$	$\Updownarrow(r1);$	$\Updownarrow(w1^h, w0);$	$T;$	$\Updownarrow(r0)\}$
ME6	ME7	ME8	ME9	ME10	ME11

This march test has twelve march elements (ME0 through ME11). The test can be derived from March S1C by eliminating the operations with the subscript b (meant to detect dirty faults) and separating the delay time T from the march element. Each three consecutive march elements in this test correspond to a single march element in March S1C (for example, ME0 in March S1C corresponds to ME0, ME1 & ME2 in March S1C$_{part}$. The test has a complexity of $(8 \cdot n + 4 \cdot h \cdot n + 4 \cdot T)$. When compared to March S1C this test replaces the highly expensive term $(4 \cdot T \cdot n)$ with the term $(4 \cdot T)$, which is constant with respect to the number of cells in the memory. This makes this test more suitable for industrial applications.

5.2 Memory layout consideration

March S2C is designed to detect two-cell soft faults. The high complexity of S2C, represented by the term $(4 \cdot n + 1)T$, is caused by the assumption that each cell in the memory can be influenced by every other cell the whole array. This assumption is not realistic. Cells are mainly influenced by their closest neighbors according to the layout of the memory.

The most widely used DRAM layout today is shown in Figure 4. The circles represent the cells in the memory, the horizontal lines represent the word lines, while the vertical lines represent the bit lines. According to this layout, each cell has three closest physical neighbors [Muhmenthaler91]. This situation is shown in the figure, where the closest neighbors of each cell are highlighted by arrows that connect between them.

Figure 4 shows how layout information can be used to accelerate soft fault testing. First, we need to select a set of memory cells that do not influence each other and consider them as **base cells** (each of which is referred to as b) for the test. The base cells are shown in Figure 4(a) as black circles. The memory cells that each base cell can influence is called its **neighborhood**, which can be denoted as $(\triangle_b)$. Then the memory test can be performed as follows:

1. Select the first set of base cells as indicated by black dots in Figure 4(a)
   (a) Initialize base cells to a specific value.
   (b) Access each base cell and its neighborhood to sensitize the fault.
   (c) Wait for idle time T.
   (d) Detect the sensitized faults.
2. Repeat the operations in item 1 for each cell in the neighborhood by considering it as a base cell in itself (as shown in Figures 4(b), (c) and (d))

In the memory organization shown in Figure 4, there are four possible sets of base cells, which means that the idle time T needed for the test can be reduced from $(4 \cdot n + 1)T$ to a constant idle time, which is independent of the number of cells in the memory. An optimized test that can detect two-cell soft faults is March C2S$_{\text{layout}}$, as shown below.

March S2C$_{layout}$ = {

$\Updownarrow_i(\Updownarrow_{b_i}(w1^h, w0), \Updownarrow_{b_i}(\Updownarrow_{\triangle_b}(w1, w0^h, r0^h)), \Updownarrow_{b_i}(r0), T, \Updownarrow_{b_i}(r0));$
ME0

$\Updownarrow_i(\Updownarrow_{b_i}(w1^h, w0), \Updownarrow_{b_i}(\Updownarrow_{\triangle_b}(w1^h, r1^h)), \quad \Updownarrow_{b_i}(r0), T, \Updownarrow_{b_i}(r0));$
ME1

$\Updownarrow_i(\Updownarrow_{b_i}(w0^h, w1), \Updownarrow_{b_i}(\Updownarrow_{\triangle_b}(w0, w1^h, r1^h)), \Updownarrow_{b_i}(r1), T, \Updownarrow_{b_i}(r1));$
ME2

$\Updownarrow_i(\Updownarrow_{b_i}(w0^h, w1), \Updownarrow_{b_i}(\Updownarrow_{\triangle_b}(w0^h, r0^h)), \quad \Updownarrow_{b_i}(r1), T, \Updownarrow_{b_i}(r1))\}$
ME3

This test has four march elements divided into two complementary parts: ME0 and ME1 versus ME2 and ME3. The test divides the memory into sets of base cells, and runs through them one by one using the parameter $i = 0, 1, ...$ In the memory example of Figure 4, there are four sets of base cells, which means that i runs from 0 to 3. ME0 in the test goes through each set of base cells using $(\Updownarrow_i)$ and starts by initializing all base cells $(\Updownarrow_{b_i})$ to 0 using $w1^h, w0$. Then, ME0 goes through every aggressor $(\Updownarrow_{\triangle_b})$ of each base cell $(\Updownarrow_{b_i})$ and sensitizes the fault in the base cell by performing $(w1, w0^h, r0^h)$. Then, a sensitizing read operation is performed on every base cell to sensitize all deceptive read disturb coupling faults $(\Updownarrow_{b_i}(r0))$, followed by idle time and a detecting read operation. ME1 has the same structure, while ME2 and ME3 are the complementary of ME0 and ME1.

The complexity of this test can be calculated as follows $(4 \cdot n(1+h) + 2 \cdot 3n(1+2h) + 2 \cdot 3n(2h) + 4 \cdot n + 4 \cdot (4T) + 4 \cdot n)$, which is equal to $(18 \cdot n + 28h \cdot n + 16T)$. This test, though very costly, has a constant idle time that is independent of the number of cells in the memory $(16T)$. This test replaces March S2C derived in Section 4 to detect all soft coupling faults.

6 Conclusions

This paper introduced new, optimized tests specifically designed to detect soft faults in DRAM devices. The opti-

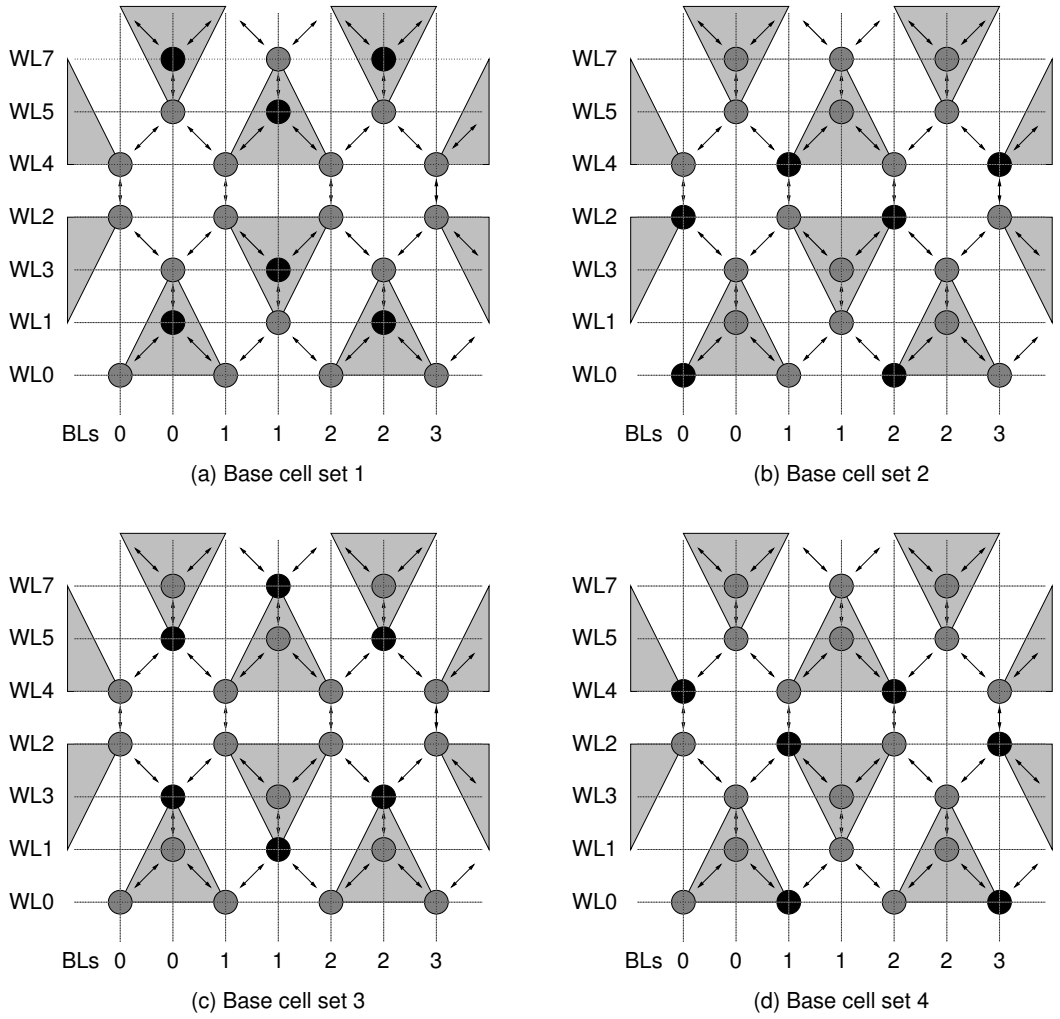


Figure 4. Sets of base cells used to accelerate soft fault testing.

mization method depends on information about the design of the memory or about its layout. Two new tests have been discussed, one that represents an optimized version for single-cell soft faults and another for two-cell faults. The complexity of both tests have been reduced in such a way that delay time does not scale with the number of cells in the memory, making them more suitable for industrial application. Specific optimization examples have been shown that enable test designers to modify the tests according to their own memory design or layout.

References

[Adams96] R.D. Adams and E.S. Cooley, "Analysis of a Deceptive Destructive Read Memory Fault Model and Recommended Testing," *in Proc. IEEE North Atlantic Test Workshop*, 1996.

[Al-Ars04] Z. Al-Ars and A.J. van de Goor, "Soft Faults and the Importance of Stresses in Memory Testing," *in Proc. Design, Automation and Test in Europe*, 2004, pp. 1084–1089.

[Al-Ars05] Z. Al-Ars, *DRAM Fault Analysis and Test Generation*, PhD thesis, Delft Univ. of Technology, Delft, the Netherlands, 2005, http://ce.et.tudelft.nl/~zaid/

[Al-Ars06] Z. Al-Ars *et al.*, "DRAM-Specific Space of Memory Tests," *in Proc. IEEE Int'l Test Conference*, 2006.

[Antonin91] G. Antonin, H.-D. Oberle and J. Kolzer, "Electrical Characterization of Megabit DRAMs, 1. External Testing," *in IEEE Design & Test of Computers*, vol. 8 , no. 3, 1991, pp. 36–43.

[Dekker90] R. Dekker, F. Beenker and L. Thijssen, "A Realistic Fault Model and Test Algorithms for Static Random Access Memories," *in IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 9, no. 6, 1990, pp. 567–572.

[Keshavarzi97] A. Keshavarzi, K. Roy and C.F. Hawkins, "Intrinsic Leakage in Low Power Deep Submicron CMOS ICs," *in Proc. IEEE Int'l Test Conf.*, 1997, pp. 146–155.

[Muhmenthaler91] H.-D. Oberle and P. Muhmenthaler, "Test Pattern Development and Evaluation for DRAMs with Fault Simulator RAMSIM," *in Proc. IEEE Int'l Test Conf.*, 1991, pp. 548–555.

[Wang01] M.-J. Wang *et al.*, "Guardband Determination for the Detection of Off-State and Junction Leakages in DRAM Testing," *in Proc. Asian Test Symposium*, 2001, pp. 151–156.