Defect Oriented Testing of the Strap Problem Under Process Variations in DRAMs

Zaid Al-Ars[1] Said Hamdioui[1] Ad J. van de Goor[1] Georg Mueller[2]

[1]Delft University of Technology
Faculty of EE, Mathematics and CS
Mekelweg 4, 2628 CD Delft, The Netherlands

[2]Qimonda AG
Department of Product Engineering
Am Campeon 12, 85579 Neubiberg, Germany

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

Abstract: *Efficient and effective methods are needed to generate defect oriented tests for todays VLSI circuits. This paper describes an industrial case study for using defect injection and Spice simulation to generate defect oriented tests for the so-called strap defects in DRAMs, taking both the sensitivity of this defect to process variations and bit line coupling into consideration. The paper discusses all the different stages of the test generation process, starting with defect modeling, followed by the simulation methodology, test generation and optimization, and finally test application and industrial evaluation performed in Qimonda. Results show that the generated tests have the same coverage as previously used tests with possible test time reduction of up to* 59%*. The analysis also identifies the slow process corner and the data backgrounds 11 and 01 as the most stressful combinations to test the strap. The paper also discusses a test method used to account for process variations and detect the fault in any process corner.*

1 Introduction

Defect oriented test generation is becoming increasingly important to derive both efficient (low-cost) and effective tests (with a high fault coverage). Simulation-based test generation has been successfully applied in both digital [8] as well as analog devices [15] to evaluate the faulty behavior and derive appropriate corresponding tests for observed faulty behavior. This approach has also been applied to memory devices to analyze cell array faults [1, 9, 13, 14, 11, 16] as well as address decoder faults [6, 7, 12] of many different types of memory.

This paper presents an industrial case study performed in Qimonda for implementing the simulation-based analysis of the faulty behavior to a specific type of open defect within memory cells: the strap problem [2]. This type of open is particularly difficult to test for, since it causes a floating voltage node in the memory cell, which makes it particularly sensitive to manufacturing process variations and *bit line (BL)* coupling effects [4]. The paper runs through the whole process of simulation-based fault analysis, starting from problem definition and modeling, through test generation, and ending with test implementation and evaluation. The main contributions of the paper are as follows:

- identifying the impact of process variations on the test process of memory devices, and generating a test to detect the faulty behavior in all process corners
- reducing the test time of memory tests for the strap problem with up to 59%
- showing the most stressful background pattern to be used as a result of BL coupling
- determining the most stressful process corner using simulation

Section 2 starts with a definition of the strap defect, along with an electrical model of it to be used for the simulation. It also discusses the concepts of process variations and process corners, and presents a way to model them at the electrical level. Section 3 identifies the DRAM specific operation sequences to be simulated. Section 4 describes the simulation-based fault analysis methodology used to analyze the faulty behavior of the strap. Section 5 derives the test pattern used to detect the strap defect and discusses the industrial evaluation results of the test. The section also shows how to take process variations into consideration, discusses their impact on the faulty behavior, and derives memory tests to detect the faults in all process corners. Finally, Section 6 ends with the conclusions.

2 Definition of the strap problem

In this section, we provide some background information regarding the open defect to be analyzed, the memory simulation model and the concept of process corners.

1-4244-4203-0/08/$20.00 2008 IEEE

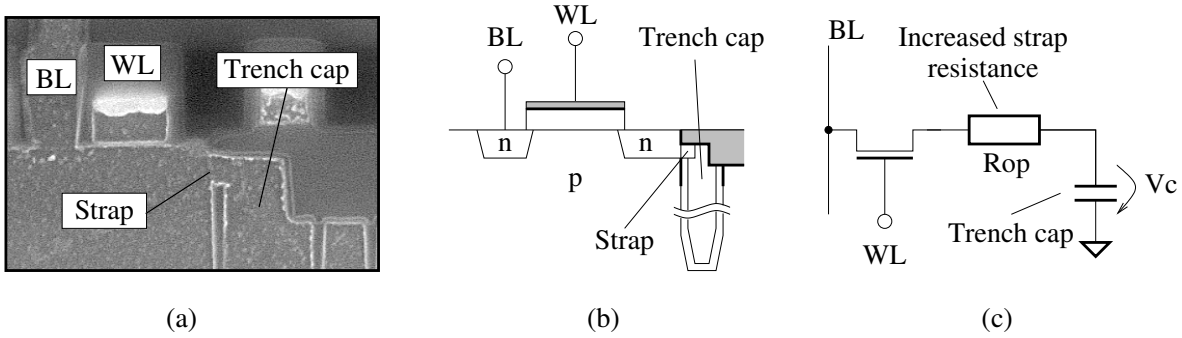


Figure 1. Modeling the cell and the strap from (a) silicon, to (b) the layout level, and (c) the electrical level.

2.1 The defect

The special open defect analyzed in this paper models an increase in the resistive value of what is called the **strap connection**. The strap connection is a conductive path between the drain region of the pass transistor of the memory cell and the trench capacitor [2]. Figure 1(a) shows a scanning electron microscope image of a DRAM memory cell, where the position of the strap is indicated. In addition, the figure show the *word line (WL)* connection, the BL connection, as well as the trench capacitor of the memory. Figure 1(b) gives a schematic representation of the cell and strap. Due to imperfections in the fabrication process, the strap may take up any resistive value according to the statistical distribution of the fabrication process.

Ideally, the memory is designed such that the strap should be manufactured with a predefined target resistance value. An *increase* in the strap resistance can be electrically modeled as an added series resistance (R_{op}) along the conductive path between the pass transistor and the trench capacitor in the cell, as shown in Figure 1(c). An increase in R_{op} reduces the ability of the memory to control the voltage stored across the cell capacitor, which leaves the stored voltage in the cell (V_c) floating to a certain extent.

From a physical point of view, the modeled increase in the strap resistance R_{op} can be attributed to a number of factors, such as a change in the doping concentration of the strap, a geometrical misalignment in the positioning or sizing of the strap, etc.

2.2 Simulation model

The memory Spice model used to perform the simulations for the strap problem is a design validation model in the 0.11μm technology, provided by Qimonda. To reduce simulation time, the memory model is reduced to include only those parts of the memory needed to perform the fault analysis. Figure 2 shows a block diagram of the memory model. The model consists of 3×2 memory cells and has three BL pairs: top true BL (BTt) and top complement BL (BCt); middle true BL (BTm) and middle complement BL (BCm); and bottom true BL (BTb) and bottom complement BL (BCb). Each BL pair is connected to 2 memory cells, one to BT and the other to BC. In addition, the model has 3 sense amplifiers (SAt, SAm and SAb), precharge circuits and access devices. A write buffer is included to enable simulating write operations, in addition to a read buffer for simulating read operations.

The model has two WLs, each is connected to three memory cells: WLt is connected to three cells on BT, while WLc is connected to three cells in BC. The fault analysis described in this paper is performed on Cellm (the memory cell connected to WLt and BTm). The behavior of cells connected to BC is the complementary to that of cells connected to BT (i.e., with all 0s replaced with 1s, and vice versa).

The top and bottom BL pairs are included in the model in order to simulate the impact of *data background (DB)* patterns on the faulty behavior [5]. Each BL is connected to the two adjacent BLs by parasitic capacitances (not shown in Figure 2). When Cellm is accessed, Cellt and Cellb are accessed at the same time (since all are connected to WLt), thereby influencing the behavior of the operations performed on Cellm. To simulate the impact of different DBs, the simulation analysis is performed for different data stored in Cellt and Cellb. There are four different DBs:

1. **DB 00**—This refers to 0s stored in both Cellt on BTt and Cellb on BTb.
2. **DB 10**—This refers to a 1 stored in Cellt on BTt and a 0 stored in Cellb on BTb.
3. **DB 11**—This refers to 1s stored in both Cellt and Cellb.
4. **DB 01**—This refers to a 0 stored in Cellt and a 1 stored in Cellb.

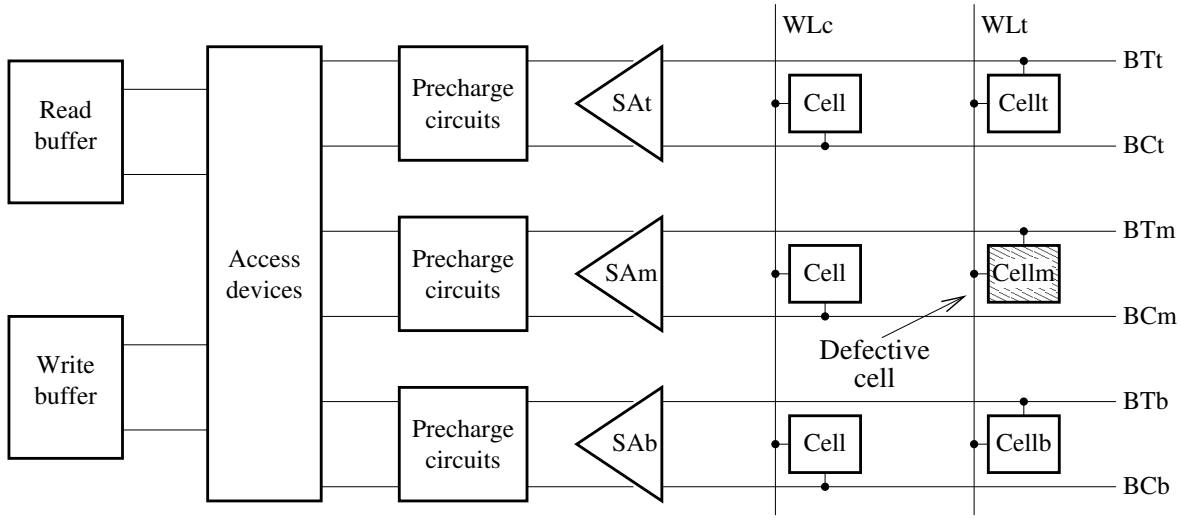


Figure 2. Spice simulation model used for defect injection and fault analysis.

2.3 Process variations

During the fabrication process of memory components, the characteristics of the produced components vary within a given range around their target specifications. These variations result in some components operating slower or faster than the targeted speed of operation. It is important to evaluate the impact of process variations on the faulty behavior of the memory, since a test should be able to detect the faulty behavior for *any* produced memory component.

Process variations can be modeled and simulated as variations in the transistor parameters used in the simulation [10]. Transistors are usually divided into two sets, PMOS and NMOS, each with independently varying parameters. Figure 3 shows how to model process variations considering PMOS vs NMOS transistor variations. The x-axis represents the measured saturation current of NMOS transistors (Id_sat NMOS), while the y-axis represents the measured saturation current of PMOS transistors (Id_sat PMOS). The ellipse represents the *process spread* around the center. Each point in the ellipse represents one possible combination of Id_sat for NMOS vs PMOS. Transistors with Id_sat at the center of the ellipse (the crossing of the two principal axes) are said to belong to the *nominal corner*. These are modeled using the nominal transistor parameters. For every other point in the ellipse, a set of variations in the transistor parameters (Δparameter) are added to the nominal parameter set.

A **process corner** is a term that refers to specific points in the ellipse of Figure 3. This means that each process corner has an associated set of transistor parameters describing transistor behavior at that point. Besides the nominal corner, the figure shows four additional process corners:

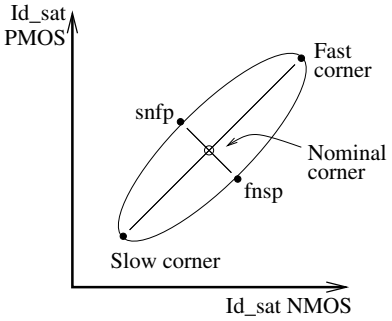


Figure 3. Modeling process variations using transistor parameters.

- **snsp corner** or slow corner—This refers to the process corner where NMOS and PMOS transistors conduct the least drain current. The symbol snsp stands for slow NMOS and slow PMOS.

- **snfp corner**—This corner has slow NMOS and fast PMOS transistors. It refers to the process corner with low NMOS drain current and high PMOS drain current.

- **fnsp corner**—This corner has fast NMOS and slow PMOS transistors. It refers to the process corner with high NMOS drain current and low PMOS drain current.

- **fnfp corner** or fast corner—This refers to the process corner where NMOS and PMOS transistors conduct the most drain current.

3 Simulated DRAM sequences

In this section, we discuss the DRAM specific operation sequences to be used for the simulation-based fault analysis approach.

3.1 DRAM commands

In order to perform the fault analysis specifically on DRAMs, there is a need to use the five DRAM commands in the simulation rather than the two generic memory operations read (r) and write (w). These commands are: Act, Rd, Wr, Pre and Nop. They are represented in Figure 4 as described next:

1. **Act**: This is the *activate* command. When this command is issued, a WL in the cell array is selected thereby a row of cells is accessed. Furthermore, an *internal* read operation is performed by moving the data from the row of memory cells to the sense amplifiers.

2. **Rd**: This is the *read* command. When this command is issued, the data in the selected sense amplifiers is moved to the data buffers and to the data bus. This resembles an *external* read operation.

3. **Wr**: This is the *write* command. When this command is issued, the data in the data buffers is moved to both the sense amplifiers and to the cell array as well. This resembles an external as well as an internal write operation.

4. **Pre**: This is the *precharge* command. When this command is issued, any selected WL is deselected and BLs are precharged.

5. **Nop**: This is the *no operation* command, which represents an idle cycle.

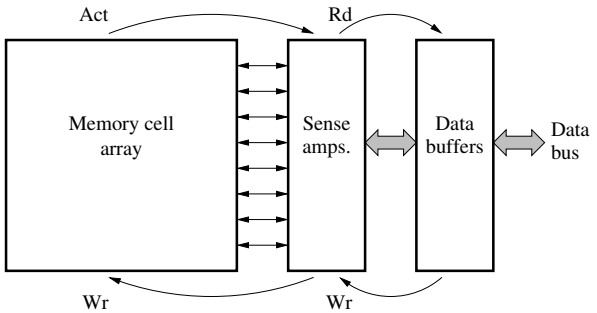


Figure 4. Functional model of a DRAM.

3.2 Simulated sequences

Since there is an infinite number of possible operation sequences to simulate, it is impossible to simulate all of them. The solution is to simulate a limited number of sequences, called *basic sequences*, and then use those to *approximate* the behavior of any other sequence. For DRAMs, basic sequences should enable us of approximating any functional sequence of the commands Act, Rd, Wr, Pre and Nop. Taking the strap problem into consideration, and since the memory cell is only accessed between the Act and Pre commands, faults can only be sensitized when the cell is accessed using the commands performed between Act and Pre. Therefore, for the analysis of the strap, we only need to simulate the following sequences:

1. Act Wrx Wrx ... Wrx Pre (to evaluate the impact of Wrx, $x \in \{0, 1\}$)

2. Act Wrx Nop Nop ... Nop Pre (to evaluate the impact of Nop after Wrx)

3. Act Rd Rd ... Rd Pre (to evaluate the impact of Rd)

4. Act Rd Nop Nop ... Nop Pre (to evaluate the impact of Nop after Rd)

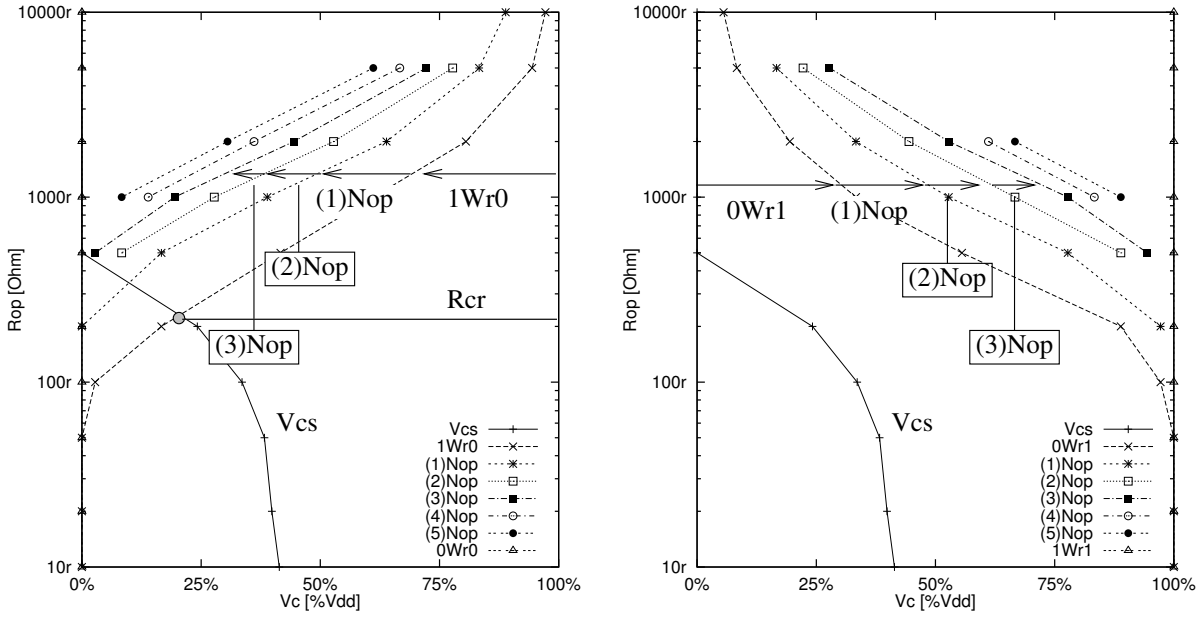
5. Act Nop Nop Nop ... Nop Pre (to evaluate the impact of Nop after Act)

Simulating these sequences enables the evaluation of the behavior of any other arbitrary sequence, such as Act Wrx Wrx Nop ... Nop Pre, or Act Wrx Nop Rdx Rdx Pre, etc. The method to do this is discussed in Section 4.

Simulations of these sequences show that (for a given value of x) Sequences 1 and 2 have exactly the same behavior. This is also true for Sequences 3, 4 and 5. Therefore, only one sequence in each group needs be simulated. We select the sequences with the most number of Nop commands as representatives, since the clock cycles with no operations can be used to test multiple memory banks in parallel, thereby reducing overall test time. This leaves us with Sequences 2 and 5.

These sequences can be reduced further. Since the impact of Act is analyzed by Sequence 5, we do not need to include it in Sequence 1. At the same time, the impact of Pre is also known, which is simply ending access to the cell and fixing the voltage within it. It is therefore not needed in any of the sequences. This leaves the following sequences chosen to analyze the faulty behavior:

- Sequence of Wr0: `Wr0 Nop Nop ... Nop`
- Sequence of Wr1: `Wr1 Nop Nop ... Nop`
- Sequence of Act: `Act Nop Nop ... Nop`



(a) Plane of Wr0

(b) Plane of Wr1

Figure 5. Result planes in the nominal corner, at nominal stress and with DB 00, for the sequences (a) Wr0, and (b) Wr1.

4 Analysis methodology

In this section, we describe the simulation-based fault analysis methodology by discussing the analysis performed at the nominal corner of the process.

4.1 Background 00

Figures 5 and 6 show the simulation results in the nominal process corner (as defined in Section 2.3), at nominal stresses (according to the specifications of the memory), and with DB 00 (0 is stored in cells on the adjacent BL pairs). The results are divided into three different result planes [3], one for each analyzed basic sequence. Each result plane describes the impact of performing successive commands on the stored voltage within the cell (V_c) for a given value of the open resistance (R_{op}), as shown in Figure 1(c). The x-axes in the result planes represents V_c, while the y-axis represents the value of R_{op}. The value of V_c is not given in absolute voltage levels, but as percentages of V_{dd}. In the same way, a scaled value of R_{op} is shown on the y-axis using the scale factor r.

Plane of Wr0: This result plane is shown in Figure 5(a). To generate this figure, the floating cell voltage V_c is initialized to the two worst case voltages, V_{dd} and GND, and then the sequence Wr0 Nop ... Nop is applied to the cell. With an initial $V_c = V_{dd}$, the sequence results in the gradual decrease (depending on the value of R_{op}) of V_c towards GND. With an initial $V_c =$ GND, the value of V_c remains at GND. The voltage level after each command in the sequence is recorded on the result plane, which results in a number of curves in the plane. All curves have names, and some of them are indicated by an arrow pointing in the direction of voltage change. The 1Wr0 curve identifies the impact of Wr0 on a cell voltage initialized to V_{dd}, while the 0Wr0 curve (the last entry in the legend) identifies the impact of Wr0 on a cell voltage initialized to GND. The curves numbered as (n)Nop indicate the impact of no operations on V_c following a 1Wr0, where n is the number of Nops needed to get to the indicated curve. The figure also shows the cell sense-threshold curve (V_{cs}), above which the sense amplifier senses a 1 and below which the sense amplifier senses a 0. The V_{cs} curve is copied from the plane of the Act sequence, which is explained in detail below [see "Plane of Act" below]. This plane enables evaluating the effect of any Wr0 on the defective cell.

Plane of Wr1: This result plane is shown in Figure 5(b). To generate this figure, V_c is initialized to the two worst case voltages V_{dd} and GND and then the sequence Wr1 Nop ... Nop is applied to the cell. With an initial $V_c =$ GND, the result is the gradual increase of V_c towards V_{dd}, while an initial V_{dd} remains as it is in the cell. The voltage

level after each command in the sequence is recorded on the result plane, which produces a number of curves in the plane. These curves are indicated in the same way as for the curves in the plane of Wr0 above. This curve enables evaluating the effect of any Wr1 on the defective cell.

Plane of Act: This result plane is shown in Figure 6. To generate this figure, first we identify the threshold voltage within the cell that determines the sense amplifier output V_{cs} (the cell voltage above which the sense amplifier detects a 1, and below which it detects a 0). Then, the sequence Act Nop ... Nop is applied twice: first for V_c that is initially marginally lower than V_{cs}, and a second time for V_c that is marginally higher than V_{cs}. After each command, V_c is recorded on the result plane, which results in a number of curves on the plane. The +Act curve indicates the impact of performing Act with V_c marginally higher than V_{cs}, while -Act indicates the impact of performing Act with V_c marginally lower than V_{cs}. The other curves indicated the impact of the nth Nop following the initial Act.This plane enables evaluating the effect of any Act on the defective cell.

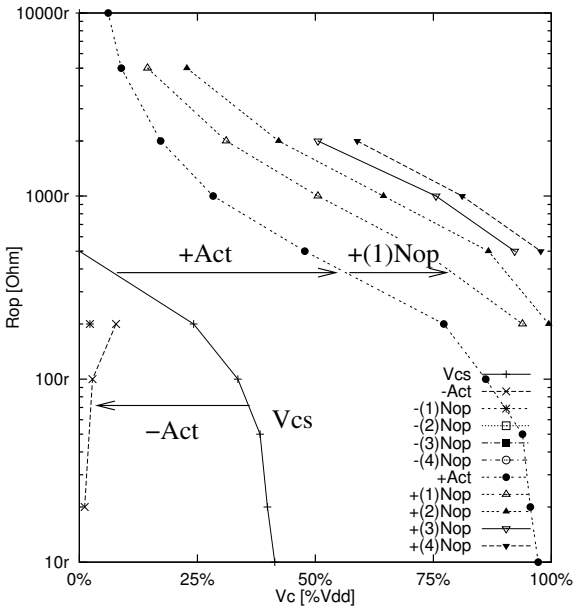


Figure 6. Result plane in the nominal corner, at nominal stress and with DB 00, for the Act sequence.

Using the result curves in Figure 5, we can analyze the following aspects of the faulty behavior:

1. Identify the *critical resistance* (R_{cr}), which is the R_{op} value where the cell *starts* to cause faults on the output, for *any* sequence of operations.
2. Generate a test that detects the faulty behavior of the defect for any resistance value and any initial floating voltage.

(1) For the fault analysis shown in Figure 5, the memory behaves properly for any operation sequence as long as $R_{op} < 210r\Omega$. To understand why, note that a fault would only be detected when a Wr1 command fails to charge V_c up above V_{cs}, or a Wr0 fails to discharge V_c to below V_{cs} (V_{cs} is indicated by a curve in Figure 5). In both situations, trying to read after performing the write would detect the faulty behavior. Note that for $R_{op} > 210r\Omega$, Wr0 fails to discharge V_c to the value needed by Act to sense a 0. This is indicated in Figure 5(a) as a dot at the intersection between the 1Wr0 curve and the V_{cs} curve. Furthermore, note that the curve 0Wr1 in Figure 5(b) does not intersect the V_{cs} curve, which means that Wr1 never fail no matter how high R_{op} becomes!

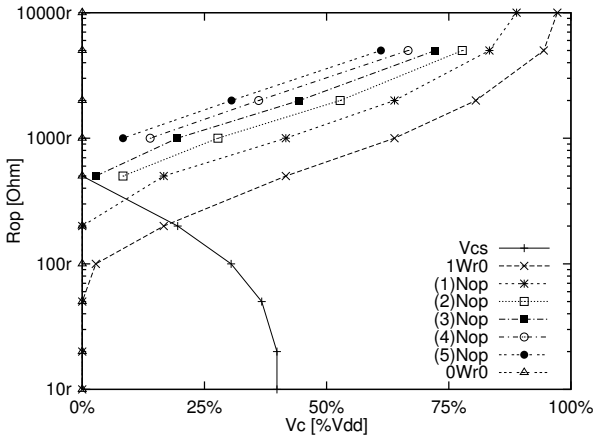
(2) Now, the result planes are used to generate a detection condition that detects the faulty behavior caused by any defect resistance for any initial floating voltage, in case a fault can be detected. Figure 5(a) shows that faults can be detected with $R_{op} \geq 210r\Omega$. Inspecting the figure shows that with $R_{op} \geq 210r\Omega$, and with any initial voltage V_c, the sequence Wr1 Nop Nop Wr0 will sensitize a fault. This can be validated by checking Figure 5(b) for $R_{op} = 210r\Omega$, and noting that performing Wr1 Nop Nop charges V_c up from any voltage (GND or higher) to approximately V_{dd}. With $V_c = V_{dd}$, performing Wr0 sensitizes the fault which can then be detected as discussed in point (1) above. Therefore to detect the fault, the detection condition $\Updownarrow$(..., Wr1, Nop, Nop, Wr0, Pre, Act, Rd0, ...) is sufficient.

4.2 Backgrounds 10, 11 and 01

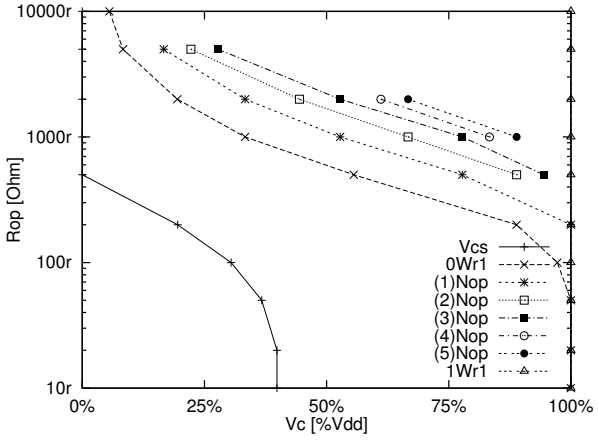
In this section, fault analysis results are given in the nominal process corner and with nominal stress conditions for DBs 10, 11 and 01. For each DB, the result planes are presented in order to discuss their faulty behavior. As the discussion in Section 4.1 have shown, the only important part of the result plane of Act is the V_{cs} curve, which is included and discussed as part of the result planes of Wr0 and Wr1. Therefore, only the result planes of Wr0 and Wr1 are discussed next, but not that of Act. A summary of the results presented here can be found in Table 1.

Background 10

Figure 7 shows the result planes for DB 10. Figures 7(a) and (b) give the results for Wr0 and Wr1, respectively. The curves in the figures show the same tendencies in the behavior as those in Figure 5.

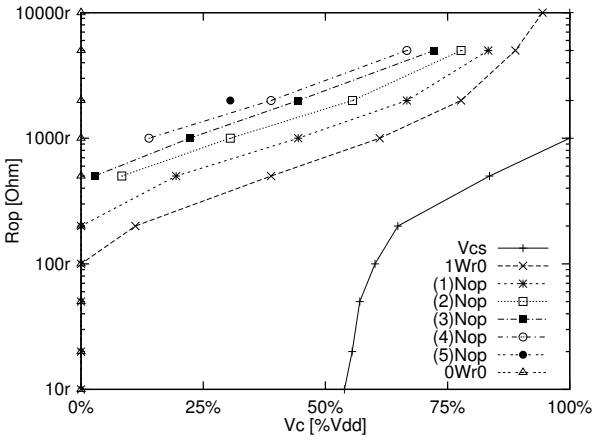


(a) Wr0

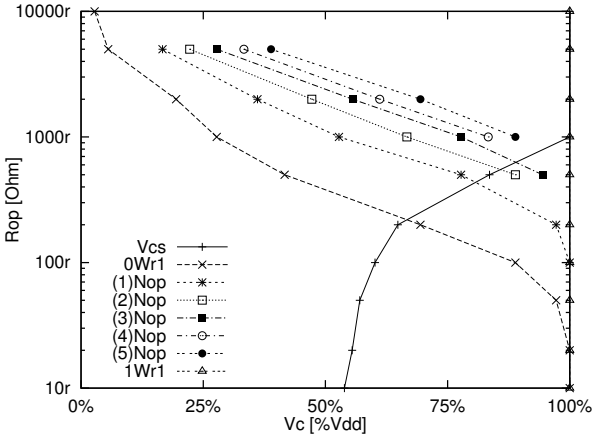


(b) Wr1

Figure 7. Result planes with DB 10 for (a) Wr0 and (b) Wr1.



(a) Wr0



(b) Wr1

Figure 8. Analysis results with DB 11 for (a) Wr0 and (b) Wr1.

Although the results in this figure are not exactly the same as those shown in Figure 5, the important aspects of the faulty behavior have not changed. The cell starts to fail when $R_{op} \geq 205r\Omega$ when the curve V_{cs} intersects 1Wr0 in Figure 7(a). This means that modifying DB from 00 to 10 does not have a big impact on the faulty behavior of the memory. Yet, the small increase in the range of failing R_{op} values indicates that DB 10 ($R_{op} \geq 205r\Omega$) is slightly more effective in stressing the test than DB 00 ($R_{op} \geq 210r\Omega$).

Background 11

Figure 8 shows the analysis results for DB 11. Figures 8(a) and (b) give the results for Wr0 and Wr1, respectively. The figures show that the V_{cs} curve changed significantly with DB 11, compared to DBs 10 and 00. The sense amplifier is now biased towards detecting a stored 0 instead of detecting a stored 1 [5].

Inspecting Figure 8(a) reveals that the V_{cs} curve does not intersect the 1Wr0 or any of the (n)Nop curves, which means that the Wr0 sequence never fails. However, the V_{cs} curve does intersect the 0Wr1 curve in Figure 8(b) at about $R_{op} = 205r\Omega$. This indicates that the Wr1 sequence starts to fail with $R_{op} \geq 205r\Omega$.

A detection condition to detect this fault is $\Updownarrow$(..., Wr0, Nop, Nop, Wr1, Pre, Act, Rd1, ...). It is interesting to note that this detection condition has a similar sequence of commands as the detection condition derived for DB 00, with the exception that the data used in this detection condition is complementary to that used in the condition for DB 00 (i.e., 1s are replaced with 0s, and vice versa).

Background 01

Simulations show that using DB 01 causes a similar faulty behavior as that caused by DB 11. However, the cell now fails when $R_{op} \geq 200r\Omega$. This means that using DB 01 slightly increases the range of failing R_{op} as compared to DB 11. In other words, it is actually easier to detect a fault

Table 1. Summary of simulation results in the 5 different process corners and for all DBs.

Corner	DB	R_{cr}	Det. condition	Corner	DB	R_{cr}	Det. condition
Nominal	00	$210r\Omega$ ■	CondT	fnfp	00	$220r\Omega$ ■	CondT
	10	$205r\Omega$ ■	CondT		10	$205r\Omega$ ■	CondT
	11	$205r\Omega$ ■	CondC		11	$220r\Omega$ ■	CondC
	01	$200r\Omega$ ■	CondC		01	$205r\Omega$ ■	CondC
snsp	00	$200r\Omega$ ■	CondT	snfp	00	$210r\Omega$ ■	CondT^{+}
	10	$185r\Omega$ ■	CondT		10	$205r\Omega$ ■	CondT^{+}
	11	$105r\Omega$ ■	CondC^{-}		11	$205r\Omega$ ■	CondC
	01	$100r\Omega$ ■	CondC^{-}		01	$200r\Omega$ ■	CondC^{-}
fnsp	00	$205r\Omega$ ■	CondT	CondT: ⇳(..., Wr1, Nop, Nop, Wr0, Pre, Act, Rd0, ...)			
	10	$200r\Omega$ ■	CondT	CondC: ⇳(..., Wr0, Nop, Nop, Wr1, Pre, Act, Rd1, ...)			
	11	$200r\Omega$ ■	CondC^{-}	CondC^{-}: ⇳(..., Wr0, Nop, Wr1, Pre, Act, Rd1, ...)			
	01	$160r\Omega$ ■	CondC^{-}	CondT^{+}: ⇳(..., Wr1, Nop, Nop, Nop, Wr0, Pre, Act, Rd0, ...)			

with DB 01 than it is with DB 11. A similar remark has been made for DBs 00 and 10.

4.3 Summary of simulation results

All 5 process corners (nominal, fnfp, snsp, snfp and fnsp) have been simulated and analyzed [see Section 2.3] at nominal stresses (nominal voltage, temperature and timing). For each process corner and each DB, the three result plains (Wr0, Wr1 and Act) have been generated. To simplify the discussion here, only the two most important outcomes of the analysis are given: (1) the critical resistance (R_{cr}) at which the memory starts to fail, and (2) the detection condition needed to detect the faulty behavior. Table 1 lists these two outcomes for each process corner. The column "Corner" lists the process corner at which the analysis is performed, the column "DB" indicates the data background, the column R_{cr} states the critical resistance, while the column "Det. condition" lists the needed detection condition.

The table identifies the following characteristics for the faulty behavior, resulting from an elevated strap resistance with nominal stresses:

1. The value of R_{cr} varies between a minimum of $100r\Omega$ for the snsp corner with DB 01, and a maximum of $220r\Omega$ for the fnfp corner with DB 00 and DB 11. This means that snsp corner with DB 01 represent the most stressful conditions for the strap problem, while the the fnfp corner with DB 00 and DB 11 represent the least stressful conditions.

2. For each process corner, DB 01 always results in the lowest R_{cr}, while DB 00 results in the highest. This means that DB 01 is the most stressful DB, while DB 00 is the most relaxed one.

3. Depending on the process corner, the difference in R_{cr} between different DBs could be as large as 100 $r\Omega$ for snsp, or as small as 10 $r\Omega$ for the nominal and the snfp corner.

4. The detection conditions needed to detect the faulty behavior caused by the defect have the same general structure for all process corners. The only difference is in the number of initializing Wr and Nop needed to precharge V_c to a strong enough voltage. This indicates that it is possible to use a single detection condition (and therefore a single march test) to detect the faulty behavior in all process corners. The only requirement is to use a large enough number of Nops to cover the worst case detection condition.

5 Test generation and evaluation

In this section, we use the detection conditions listed in Table 1 to generate a march test to detect the fault, taking into account the needed DB pattern.

5.1 Simulation-based tests

According to Table 1, DBs 00 and 10 share the same worst-case detection condition corresponding to the corner snfp, which is ⇳(..., Wr1, Nop, Nop, Nop, Wr0, Pre, Act, Rd0, ...). The fact that changing the value of Cellt between 0 and 1 does not change the detection condition means that Cellt is insignificant for the faulty behavior. This indicates that

in order for this detection condition to work, only Cellb [see Figure 2] should contain a 0. This is done using a memory initialization step to write a 0 in the whole memory, as follows ⇳(Act, Wr0, Pre).

For DBs 11 and 01, the worst-case detection condition is ⇳(..., Wr0, Nop, Nop, Nop, Wr1, Pre, Act, Rd1, ...). In this case, the DB indicates that only a value of 1 contained in Cellb is significant for the detection condition. This is done using an initialization step to write a 1 in the whole memory, as follows ⇳(Act, Wr1, Pre).

Therefore, there are two possible tests, any of which detects the faulty behavior. In order to make the operations in the detection conditions complete, each operation has to start with an Act and has to end with a Pre. The subscripts 00 and 11 stand for the DB used in the test.

1. T_{00} = {⇳(Act, Wr0, Pre);
   ⇳(Act, Wr1, Nop, Nop, Nop, Wr0, Pre,
   Act, Rd0, Pre)}

2. T_{11} = {⇳(Act, Wr1, Pre);
   ⇳(Act, Wr0, Nop, Nop, Nop, Wr1, Pre,
   Act, Rd1, Pre)}

Since each of the two tests detects the faulty behavior, we only need to perform one of them to properly test for the strap problem. Referring to Table 1, it is clear that T_{11} has a lower R_{cr} than T_{00} in all analyzed process corners. Therefore, it is recommended to use T_{11}.

5.2 Practical aspects

If we assume that the memory functions exactly as the electrical model does on a simulator, the generated tests above would function exactly as expected. Practically, however, real silicon may deviate from the simulated behavior. As a result, the exact number of Nops needed to charge the memory to the desired voltage may differ. It is possible to ensure a higher fault coverage even for memories that deviate from the simulation model by increasing the number of Nops in the test. This gives the following list of tests:

- Original test: T_{00} =
  {⇳(Act, Wr0, Pre);
  ⇳(Act, Wr1, Nop, Nop, Nop, Wr0, Pre,
  Act, Rd0, Pre)}

- Add 1 Nop: $T_{00}{}^{+}$ =
  {⇳(Act, Wr0, Pre);
  ⇳(Act, Wr1, Nop, Nop, Nop, **Nop,** Wr0, Pre,
  Act, Rd0, Pre)}

- Add 2 Nops: $T_{00}{}^{++}$ =
  {⇳(Act, Wr0, Pre);
  ⇳(Act, Wr1, Nop, Nop, Nop, **Nop, Nop,** Wr0, Pre,
  Act, Rd0, Pre)}

- etc.

Keep in mind that the more Nops are added, the less the impact each additional Nop will have, because the closer the cell voltage gets to V_{dd} the more difficult it becomes to charge the cell higher.

5.3 Common industrial tests

The following test is commonly used for detecting an increased strap resistance [4, 17]:

T_{ind} = {⇳(Act, Wr0, Pre); M0
⇳(Act, Wr1, Pre,
Act, Rd1, Nop, Nop, Nop, Wr0, Pre,
Act, Rd0, Pre); M1
⇳(Act, Wr1, Pre); M2
⇳(Act, Wr0, Pre,
Act, Rd0, Nop, Nop, Nop, Wr1, Pre,
Act, Rd1, Pre)} M3

The march element M0 takes care of initializing the whole memory to an initial 0 state; M0 is identical to the first march element of T_{00}. M1 starts with a Wr1 to charge the cell up, which is done in the same way by T_{00}. The test then checks the stored value in the cell using a Rd1 to ensure that V_{dd} has been reached, followed by three Nops to ensure full restoration of the stored 1, and then a Wr0 is performed to sensitize the fault. Finally, a read operation detects whether a fault is sensitized. March elements M2 and M3 have the same sequence as M0 and M1, respectively, but apply the complementary data to the memory; this part of the test corresponds to T_{11}.

This test applies a modified version of both T_{00} and T_{11}. The modification involves an added sequence of Pre Act Rd in M1 and M3. According to the fault analysis performed above, there is no clear benefit for this added Rd in detecting the strap problem. In addition, since T_{11} always has a higher coverage than T_{00}, it is sufficient to perform T_{11} instead of both.

In conclusion, the analysis performed in this chapter recommends reducing the industrially used test with 32 commands to the much reduced test T_{11} with 13 commands, which reduces the test time by 59%.

5.4 Industrial evaluation

An experimental version of the strap test has been included into the test flow of a commodity DRAM product in Qimonda in order to evaluate the effectiveness of the tests proposed by our analysis as opposed to the commonly used

industrial test for the strap problem (T_{ind}). The experimental version of our proposed test for the strap employs the following sequence:

T_{exp} = {⇕(Act, Wr0, Pre); M0
⇕(Act, Wr1, Nop, Nop, Nop, Wr0, Pre,
Act, Rd0, Pre); M1
⇕(Act, Wr1, Pre); M2
⇕(Act, Wr0, Nop, Nop, Nop, Wr1, Pre,
Act, Rd1, Pre)} M3

T_{exp} is similar to T_{ind}, but with the sequence "Pre, Act, Rd1" removed from M1 and M3. T_{exp} is made up of the concatenation of the two tests T_{00} and T_{11}. As mentioned earlier in this section, it is sufficient in principle to use T_{11} to detect all cells with a strap problem, as long as the physical high voltage V_{dd} is used to represent the background DB 11. Although it is possible to write physical data for most of the cells in the cell array, it is not possible to do this for parts of the cell array where failing BL pairs are repaired with redundant elements. Therefore, it is necessary to test for both DB 00 and DB 11, to insure that the most stressful background is used.

T_{ind} and T_{exp} have been included into the test flow in Qimonda for months, and the detection results of both tests have been compared. The comparison shows that the coverage of both tests is identical for those failing cells diagnosed subsequently by failure analysis to suffer exclusively from an increased strap resistance. This validates the effectiveness of the newly proposed test, and shows the capability of the simulation-based fault analysis method to derive industrial grade memory tests.

6 Conclusions

This paper presented a case study to apply the simulation-based fault analysis method in analyzing the faulty behavior of the elevated strap problem. The analysis results make it possible to generate test patterns to detect the faulty behavior, and to ensure that these test function properly despite manufacturing process variations and coupling effects. The paper discusses the application of the analysis in detail, starting from the stage of defining an electrical model used in the simulation, through test derivation, till the stage of test application. The analysis shows that using the data backgrounds 11 and 01 in the slow process corner (snsp corner) creates the most stressful combination to test for the strap defect. It is also shown that the effects of process variations can be accounted for in the test by adding extra Nop commands within the initializing write operation (i.e., giving the cell more time to be charged during initialization). The industrial evaluation of the proposed tests indicates that they have the same coverage as other industrially derived tests, but with a test time reduction of up to 59%.

References

[1] 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.

[2] E. Adler *et al.*, "The Evolution of IBM CMOS DRAM Technology," *in IBM J. of Research and Development*, vol. 39, no. 1–2, 1995, pp. 167–188.

[3] Z. Al-Ars and A.J. van de Goor, "Approximating Infinite Dynamic Behavior for DRAM Cell Defects," *in Proc. IEEE VLSI Test Symp.*, 2002, pp. 401–406.

[4] 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/books/phd.html

[5] Z. Al-Ars, S. Hamdioui, A.J. van de Goor and S. Al-Harbi, "Influence of Bit Line Coupling and Twisting on the Faulty Behavior of DRAMs," *in IEEE Trans. on Computer-Aided Design*, vol. 25, no. 12, 2006, pp. 2989–2996.

[6] M. Azimane and A.K. Majhi, "New Test Methodology for Resistive Open Defect Detection in Memory Address Decoders," *in Proc. IEEE VLSI Test Symp.*, 2004, pp. 123–128.

[7] S. Borri *et al.*, "Defect-Oriented Dynamic Fault Models for Embedded-SRAMs," *in Proc. European Test Workshop*, 2003, pp. 23–28.

[8] P. Camurati, F. Corno, P. Prinetto and M. Sonza Reorda, "A Simulation-Based Approach to Test Pattern Generation for Synchronous Sequential Circuits," *in Proc. IEEE VLSI Test Symp.*, 1992, pp. 263–267.

[9] R. Dekker F. Beenker and L. Thijssen, "A Realistic Fault Model and Test Algorithms for Static Random Access Memories," *in IEEE Trans. on Computer-Aided Design*, vol. 9, no. 6, 1990, pp. 567–572.

[10] D. Foty, *MOSFET Modeling with Spice, Principles and Practice*, Prentice Hall Inc., New Jersey, 1997.

[11] O. Ginez *et al.*, Embedded Flash Testing: Overview and Perspectives," *in Proc. Design and Test of Integrated Systems in Nanoscale Technology*, 2006, pp. 210–215.

[12] S. Hamdioui, Z. Al-Ars and A.J. van de Goor, "Opens and Delay Faults in CMOS RAM Address Decoders," *in IEEE Trans. on Computers*, vol. 55, no. 12, 2006, pp. 1630–1639.

[13] K.-J. Lin and C.-W. Wu, "Testing Content-Addressable Memories Using Functional Fault Models and March-Like Algorithms," *in IEEE Trans. Computer-Aided Design*, vol. 19, no. 5, pp.577–588.

[14] M.G. Mohammad and K.K. Saluja, "Simulating Program Disturb Faults in Flash Memories Using SPICE Compatible Electrical Model," *in IEEE Trans. Electron Devices*, vol. 50, no. 11, 2003, pp. 2286–2291.

[15] N. Nagi and J. Abraham, "Hierarchical Fault Modeling for Linear Analog Circuits," *in J. Analog Integrated Circuits and Signal Processing*, vol. 10, no. 1–2, 1996, pp. 89–99.

[16] S. Naik, F. Agricola and W. Maly, "Failure Analysis of High Density CMOS SRAMs," *in IEEE Design and Test of Computers*, vol. 10, no. 2, 1993, pp. 13–23.

[17] J.E. Vollrath, "Testing and characterization of SDRAMs," *in IEEE Design & Test of Computers*, vol. 20, no. 1, 2003, pp. 42–50.