Approximating Infinite Dynamic Behavior for DRAM Cell Defects

Zaid Al-Ars   Ad J. van de Goor
Section of Computer Engineering, Faculty of Information Technology and Systems
Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
E-mail: z.e.al-ars@its.tudelft.nl

Abstract: *Analyzing the dynamic faulty behavior in DRAMs is a severely time consuming task, because of the exponential growth of the analysis time needed with each memory operation added to the sensitizing operation sequence of the fault. In this paper, a new fault analysis approach for DRAM cell defects is presented where the total infinite space of dynamic faulty behavior can be approximated within a limited amount of analysis time. The paper also presents the analysis results for some cell defects using the new approach, in combination with detection conditions that guarantee the detection of any detectable dynamic faults in the defective cell.*

Key words: *infinite dynamic faults, DRAMs, functional fault models, defect simulation, memory testing.*

1 Introduction

Any memory faulty behavior can be denoted by the following notation $<S/F/R>$, referred to as a *fault primitive (FP)* [vdGoor00]. S describes the *sensitizing operation sequence (SOS)* that sensitizes the fault; F describes the value of the faulty cell, $F \in \{0, 1\}$; and R describes the logic output level of a read operation, $R \in \{0, 1, -\}$. If the operation that sensitizes the fault is a write operation, then the memory has no expected output, in which case R will be assigned the value '–'. For example, the up-transition fault (TF↑) is denoted by $<0w1/0/->$, the down-transition fault is denoted by $<1w0/1/->$, and the read destructive 1 fault (RDF$_1$) is denoted by $<1r1/0/0>$. FPs can be classified according to $\#C$, the number of different cells accessed during an SOS, and according to $\#O$, the number of different operations performed in an SOS. A taxonomy of FPs is shown in Figure 1.

Any increase in $\#O$ translates into *an exponential* increase in the number of SOSes and FPs to be analyzed, which in turn exponentially increases the simulation time. This can be clearly seen in the following relations for the number of single-cell SOSes and single-cell FPs ($\#C = 1$) as a function of $\#O$ [Al-Ars99]:

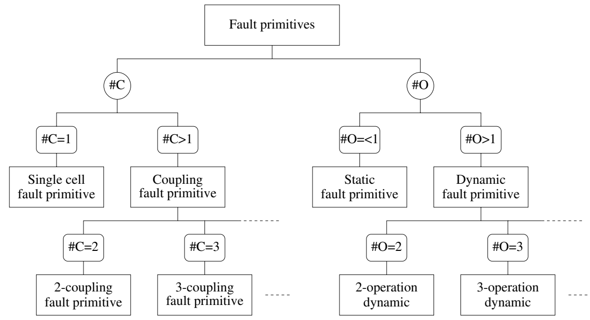


Figure 1. Taxonomy of fault primitives.

$$\#\text{single-cell SOSes} = 2 \cdot 3^{\#O}$$

$$\#\text{single-cell FPs} = \begin{cases} 2 & : \#O = 0 \\ 10 \cdot 3^{(\#O-1)} & : \#O \geq 1 \end{cases}$$

In the first relation, note that any SOS starts with either 0 or 1, followed by either $w0$, $w1$ or r. In the second relation, two state faults are possible when $\#O = 0$, 10 faults are possible (2 for $w0$, 2 for $w1$, and 6 for r) when $\#O = 1$, which is multiplied by 3 for each added operation.

Figure 1 indicates the presence of two different types of faults: *static faults* which refer to faults sensitized by at least one single operation, and *dynamic faults* which refer to faults sensitized by more than one operation [vdGoor00]. The analysis of dynamic faulty behavior of memories is inherently difficult since it requires predicting the behavior of an unlimited (infinite) number of sequences of operations. This is also true when simulation is used to establish the dynamic behavior, since such simulations require an unlimited amount of time to perform.

Previous work on dynamic behavior has either been limited to the impact of specific types of memory operations (sequences of reads, for example) [vdGoor98], or only concerned with analyzing a limited number of dynamic sequences to limit simulation time [Al-Ars00, Al-Ars01]. In this paper, we introduce a new approach where the total (infinite) dynamic faulty behavior for a given defect can be approximated. Furthermore, it is shown that the new analysis is relatively faster than the conventional one.

This paper begins with a description of the conventional approach to analyze the faulty behavior in Section 2. Section 3 introduces the new approximation approach to perform the fault analysis. Then, Section 4 presents the results of applying the new analysis on a memory using defect injection and simulation. Finally, Section 5 ends with the conclusions.

2 Conventional analysis

In this section, the conventional fault analysis approach, called the *precise simulation*, is discussed. The section starts with an example, then the properties of the precise simulation are presented.

2.1 Example of analysis

Consider the open defect (R_{op}) within a DRAM cell, as shown in Figure 2, where Spice simulations are to be used to analyze the faulty behavior resulting from this open. The analysis takes a range of possible open resistances into consideration ($1\Omega \leq R_{op} \leq 10$ MΩ, for example). The injected open in the cell model creates a floating node of the cell capacitor, the voltage of which (V_c) may vary between $\mathrm{V_{dd}}$ = 2.4 V and GND. Determining which of the two sides of an injected open is floating depends on the type of the open [Al-Ars02]. The floating node for opens within memory cells is taken to be the node connected to the cell capacitor, since the other node is controlled by the pass transistor.

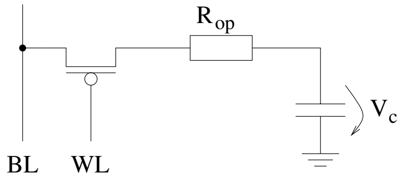


Figure 2. DRAM memory cell with an open defect.

Next, the fault analysis is performed for some points in the (V_c, R_{op}) plane, which is called the *analysis space*. Therefore, a number of values for V_c and R_{op} are selected to perform the fault analysis. This usually corresponds to applying a grid on the analysis space giving rise to a number of intersection points where the analysis is performed.

For each analysis point in the analysis space , the faulty behavior of the memory is analyzed by simulating a number of memory operations. The more operations are simulated, the more accurate our understanding of the faulty behavior becomes. However, the number of different possible SOSes grows exponentially with respect to $\#O$ according to the relation: $\#\text{SOSes} = 2 \cdot 3^{\#O}$. Therefore, the more operations are performed, the more time it takes to carry out the analysis, which makes it important to limit the number of used operations. For example, if sequences of only two operations ($\#O = 2$) are considered to be performed on a single memory cell, then $2 \cdot 3^{\#O} = 18$ different sequences of $w0$, $w1$ and r are possible. Each of these sequences has to be performed at each analysis point in the analysis space.

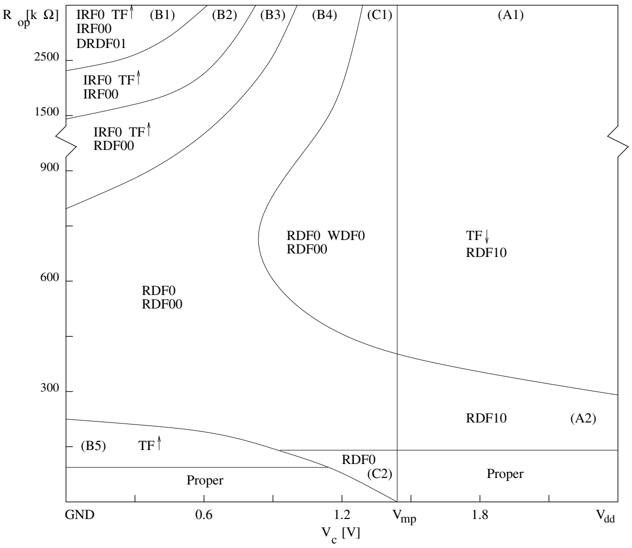


Figure 3. Conventional fault analysis results of the open in Figure 2 in the (V_c, R_{op}) analysis space.

Figure 3 shows the fault analysis results performed for the open shown in Figure 2. The results were generated SOSes of at most 2 operations [Al-Ars01]. The analysis results are organized as fault regions in the analysis space, and they change gradually (i.e., continuously) with respect to V_c and R_{op}. For example, Region B5 in Figure 3 contains the fault TF↑ ($<0w1/0/->$), while Region A1 contains the static fault TF↓ ($<1w0/1/->$) and the dynamic fault RDF_{10} ($<1w0r0/1/1>$).

2.2 Fault analysis time

In this section, we will try to estimate the time needed to perform the fault analysis using the precise simulation approach. The time needed can be described by the following relation:

$$T_{psim} = \#P \cdot \#\text{SOS} \cdot T_{sos}$$

where $\#P$ is the number of analysis points in the analysis space, $\#$SOS is the number of SOSes to be performed for each analysis point (equals $2 \cdot 3^{\#O}$), and T_{sos} is the time needed to simulate each SOS. Furthermore, $\#P$ can be further decomposed into $\#P = \#X \cdot \#Y$, where $\#X$ is the number of points taken along the x-axis of the analysis space, and $\#Y$ is the number of points taken along the y-axis of the analysis space. T_{sos} can also be further decomposed as $T_{sos} = T_o \cdot \#O$, where T_o is the simulation time needed for a single memory operation.

We use the analysis performed in Figure 3 as an example, where V_c is taken to be the x-axis and R_{op} is taken to be the y-axis.

1. $\#X$ = 10 points (10 V_c values on a linear scale, GND $\leq V_c \leq$ 2.4 V)
2. $\#Y$ = 16 points (2 R_{op} values per decade on a logarithmic scale, $1\Omega \leq R_{op} \leq$ 10 MΩ)
3. #SOS = 18 (2-operation SOSes)
4. T_o = 10 s
5. $\#O$ = 2

This adds up to $T_{psim} = \#X \cdot \#Y \cdot \#\text{SOS} \cdot T_o \cdot \#O = 10 \cdot 16 \cdot 18 \cdot 10 \cdot 2 = 57600$ s = 16 hours. Note that despite the restriction of the analyzed $\#O$ to 2, the simulation still takes a long time to perform.

3 New analysis

In this section, the new fault analysis approach, called the *approximate simulation*, is discussed. The section starts with an example, then the properties of the approximate simulation approach are presented.

3.1 Example of analysis

The new approximate simulation approach is different from precise simulation in that it enables investigating the analysis space for operation sequences with *any* $\#O$, but with a limited amount of analysis time. It achieves this by compromising the accuracy of the results.

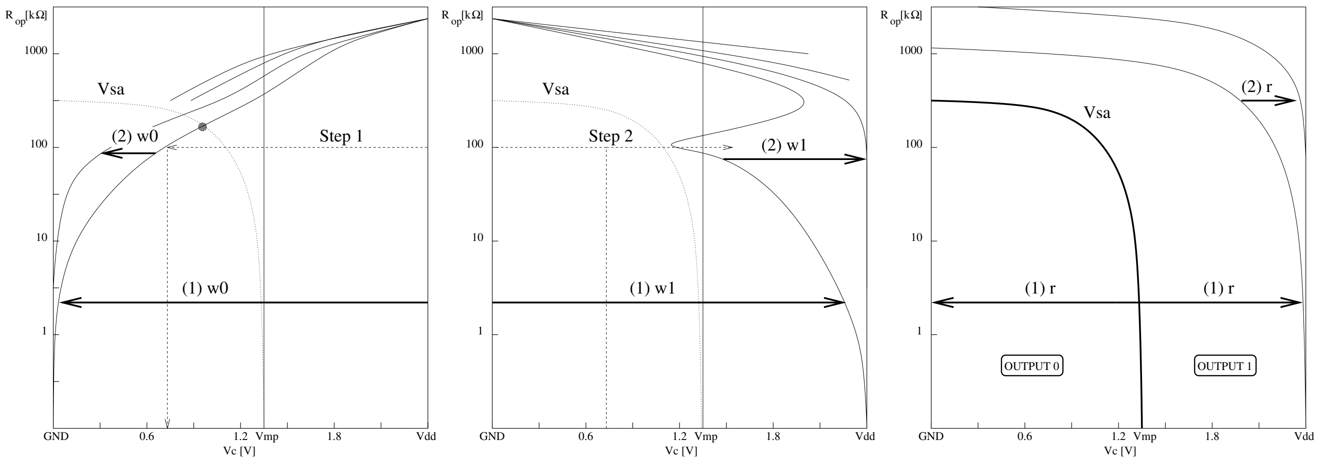
Consider the defective DRAM cell shown in Figure 2, where an open (R_{op}) makes the voltage across the cell capacitor (V_c) relatively floating. The analysis takes a range of possible open resistances ($1\Omega \leq R_{op} \leq$ 10 MΩ) and possible cell voltages (GND $\leq V_c \leq$ V$_{\text{dd}}$) into consideration.

Next, a number of R_{op} values are selected for which the analysis is to be performed. In this approach, three different (V_c, R_{op}) result planes are generated, one for each memory operation ($w0$, $w1$, and r). Each result plane describes the impact of successive $w0$, successive $w1$, or successive r operations on V_c for a given value of R_{op}. Figure 4 shows the three result planes for the three memory operations performed for the open shown in Figure 2.

Plane of $w0$: This result plane is shown in Figure 4(a). To generate this figure, the floating cell voltage V_c is initialized to V$_{\text{dd}}$ (because a $w0$ operation is performed) and then the operation sequence $1w0w0...w0$ is applied to the cell for each simulated value of R_{op}. The net result of this sequence is the gradual decrease (depending on the value of R_{op}) of V_c towards GND. The voltage level after each $w0$ operation is recorded on the result plane, which results in a number of curves in the plane. Each curve is indicated by an arrow pointing in the direction of voltage change. The arrows are numbered as $(n)w0$ where n is the number of $w0$ operations needed to get to the indicated curve. We stop performing the $w0$ sequence when the voltage change ΔV_c as a result of $w0$ operations becomes small enough. In this example, we stop the sequence when $\Delta V_c \leq$ 0.24 V since simulations show that this value is small enough to give a good resolution of the faulty behavior. The midpoint voltage (V_{mp}) (the cell voltage that makes up the border between a stored 0 and 1) is also indicated in the figure with a solid vertical line. The figure also shows a dotted curve for the sense amplifier threshold voltage (V_{sa}). This curve is discussed below when the result plane of r is presented.

Plane of $w1$: This result plane is shown in Figure 4(b). To generate this figure, V_c is initialized to GND and then the operation sequence $0w1w1...w1$ is applied to the cell. The result is the gradual increase of V_c towards V$_{\text{dd}}$. The voltage level after each $w1$ operation is recorded on the result plane, which gives a number of curves in the plane. The curves are indicated in the same way as for the curves in the plane of $w0$. We stop the $w1$ sequence when ΔV_c becomes small enough (0.24 V in this example). It is interesting to note the bump the curve $(1)w1$ has in Figure 4(b) at about $R_{op} = 100$ kΩ. This is the R_{op} value above which the sense amplifier fails in sensing the stored 0, and senses a 1 instead, which helps the $w1$ operation in charging up the cell to a higher V_c. V_{mp} is also indicated in the figure using a solid vertical line, and V_{sa} as a dotted curve.

Plane of r: This result plane is shown in Figure 4(c). To generate this figure, first the threshold voltage of the sense amplifier (V_{sa}) (the cell voltage above which the sense amplifier detects a 1, and below which detects a 0) is established and indicated on the result plane (shown as a bold curve in the figure). The curve bends gradually towards GND as R_{op} increases and the voltage differential across the sense amplifier decreases. This means that the sense amplifier in the model is not perfectly balanced, but is slightly biased towards detecting a 1 on the expense of detecting a 0. Then the sequence $rrr...r$ is applied twice: first for V_c that is initially marginally less than V_{sa}, and a second time for V_c that is marginally more than V_{sa}. In



(a) plane for w0

(b) plane for w1

(c) plane for r

Figure 4. Result planes of the approximate simulation for the operations (a) $w0$, (b) $w1$, and (c) r.

our example, we perform the sequence after initializing the cell to a V_c that is 0.12 V higher and 0.12 V lower than V_{sa}. This voltage is half of the ΔV_c we chose to stop our simulations. The voltage level after each r operation is recorded on the result plane which results in a number of curves on the plane. The curves are indicated in the same way as for the curves in the plane of $w0$. For example, with $V_c < V_{sa}$ (the part below the bold curve in the figure) only one operation is enough to set GND in the cell.

3.2 Approximating the behavior

It is possible to use the result planes of Figure 4 to analyze a number of aspects of the faulty behavior. We mention three aspects here and show how to derive them from the figure.

1. Predict faults in any fault region of the conventional precise simulation and of any operation sequence.

2. Indicate the R_{op} value where the cell *starts* to cause faults on the output for any sequence of operations.

3. Generate a test that detects the faulty behavior of the defect for any resistance value and any initial floating voltage.

(1) It is possible to predict any fault region observed by the precise simulation shown in Figure 3. For example, Region B5 in Figure 3 with the fault TF↑ can be anticipated from Figure 4(b) that describes the sequence of $w1$ operations on V_c. The curve of $(1)w1$ starts at V_{dd} for low R_{op} values, then it decreases fast and becomes lower than V_{mp} around $R_{op} \approx 100$ kΩ, only for a small range of R_{op} values. At this point, the $w1$ operation fails to set a high enough voltage within the cell and TF↑ is observed. This shows that it is possible use the approximate analysis to derive the faulty behavior of precise analysis.

The three result planes can also be used to approximate the faulty behavior of *any sequence* of memory operations. For example, the results shown in Figure 4 can be used to find out the behavior of, say, $1w0w1r1$ for $R_{op} = 100$ kΩ. The behavior is evaluated as follows:

1. Starting with an initial V_c = V_{dd} = 2.4 V, check the value of V_c after performing one $w0$ operation, (V_c = V_{dd}) $\xrightarrow{w0}$(V_c = 0.7 V); see dashed line of Step 1 in Figure 4(a).

2. Using the new $V_c = 0.7$ V, check the value of V_c after performing one $w1$ operation, (V_c = 0.7 V) $\xrightarrow{w1}$($V_c >$ 1.15 V); see dashed line of Step 2 in Figure 4(b). The figure shows that starting with V_c = GND, one $w1$ operation pulls V_c up to 1.15 V. This means that starting with 0.7 V $>$ GND in the cell, a $w1$ operation should pull V_c up to at least 1.15 V.

3. Using $V_c > 1.15$ V, check the behavior of the read operation, ($V_c > 1.15$ V) $\xrightarrow{r}$($V_c > 2.1$ V), output = 1.

This means that the memory behaves properly and no fault is detected using the sequence $1w0w1r1$ for $R_{op} = 100$ kΩ.

(2) The approximate simulation can also be used to state the R_{op} value below which the memory behaves properly for *any* possible operation sequence. For the fault analysis shown in Figure 4, the memory would behave properly for any operation sequence as long as $R_{op} < 200$ kΩ. To understand why, note that a fault would only be detected when a $w1$ operation fails to charge V_c up above V_{sa}, or a $w0$ fails to discharge V_c to below V_{sa}, where

V_{sa} is indicated by the dotted curve in Figures 4(a) and (b) and the bold line in (c). In both situations, performing a r after the w would detect the faulty behavior. Note that for $R_{op} > 200$ kΩ, $w0$ fails to discharge V_c to the value needed by r to detect a 0. This is indicated in Figure 4(a) as a dot at the intersection between the $(1)w0$ curve and V_{sa} curve. Note that the curve $(1)w1$ in Figure 4(b) does not intersect the V_{sa} curve, which means that $w1$ operations can never result in detecting a fault.

(3) The approximate simulation can also be used to generate a test that detects the faulty behavior caused by any defect resistance for any initial floating voltage, in case a fault can be detected. In the case of Figure 4, faults can be detected with $R_{op} \geq 200$ kΩ. Inspecting the figure shows that with $R_{op} \geq 200$ kΩ, and with any voltage V_c, the sequence $w1w1w0r0$ will detect a fault. For $R_{op} = 200$ kΩ, this can be validated by noting that performing two $w1$ operations charges V_c up from any voltage (GND or higher) to V$_{\text{dd}}$. With V_c = V$_{\text{dd}}$, the sequence $w0r0$ detects a fault as discussed in point (2) above. Therefore, the detection condition $\Updownarrow (..., w1, w1, w0, r0, ...)$ detects any faulty behavior for R_{op}.

3.3 Fault analysis time

The approximate simulation is much less time consuming than the precise simulation. The time needed can be described by the following relation:

$$T_{asim} = \#P \cdot \#\text{SOS} \cdot T_{sos}$$

where $\#P$ is the number of analysis points in the analysis space, #SOS is the number of SOSes to be performed for each analysis point, and T_{sos} is the time needed to simulate each SOS. In the approximate simulation, #SOS = 3 since we use only 3 SOSes, a sequence of $w0$, $w1$ and r. Furthermore, $\#P = \#Y$, where $\#Y$ is the number of points taken along the y-axis of the analysis space ($\#X$ is dropped since we do not take any point on the x-axis). T_{sos} can be further decomposed as $T_{sos} = T_o \cdot \#O$, where T_o is the simulation time needed for a single memory operation.

The $\#O$ for the approximate simulation depends on how fast a given SOS charges the memory cell. However, in order to keep a simulation accuracy along the x-axis that is approximately as good as that of the precise simulation, we need at most $\#X$ points along the x-axis, which means that we need at most $\#O = \#X$. Operations, however, usually charge V_c fast for most of the R_{op} range, thereby reducing the average $\#O$ needed.

We use the analysis performed in Figure 4 as an example, where V_c is taken to be the x-axis while R_{op} is taken to be the y-axis.

1. $\#Y$ = 16 points (2 R_{op} values per decade on a logarithmic scale)
2. #SOS = 3 ($w0$, $w1$ and r SOSes)
3. T_o = 10 s
4. $\#O \approx 4$ (this is the average over the R_{op} range)

This adds up to $T_{asim} = \#Y \cdot \#\text{SOS} \cdot T_o \cdot \#O = 16 \cdot 3 \cdot 10 \cdot 4 = 1920$ s = 0.53 hours, which is about 30 times faster than precise simulation. The difference can be much higher if the number of operations increases. The general theoretical worst case speedup of the approximate simulation approach can be given by:

$$\boxed{\text{Speedup} = \frac{T_{psim}}{T_{asim}} = 2 \cdot \#O \cdot 3^{\#O-1}}$$

which is an exponential speedup with respect to $\#O$.

4 Application results

The approximate approach has been applied to analyze the faulty behavior of a number of DRAM cell defects. This section presents the simulation methodology first, then the application results are discussed.

4.1 simulation methodology

The used electrical simulation model is a simplified design-validation model of a real DRAM. The simplified model includes one folded cell array column (2×2 memory cells, 2 reference cells, precharge devices and a sense amplifier), one write driver and one data output buffer. The used simulation tool is Pstar which is a Spice like simulation program.

Figure 5 shows the simulated defects. In total, there are 7 analyzed defects: 3 opens, 2 shorts and 2 bridges. Opens are added resistive components on signal lines within memory cells. Shorts are resistive connections to V$_{\text{dd}}$ or GND. Bridges are resistive connections between nodes within the memory cell. The simulated defect resistance values are taken in the range (1 k$\Omega \leq R \leq 10$ MΩ) on a logarithmic scale. In general, two values are simulated per decade, but more points are used for interesting regions.

For all defects, the cell voltage (V_c) has been used as the floating node voltage in the analysis. For defects within cells, all cell array voltages, other than V_c, are normalized to proper initial voltages by the beginning of each memory operation.

4.2 Simulation results

Table 1 summarizes the simulation results. Two pieces of information are given for each simulated defect. The col-

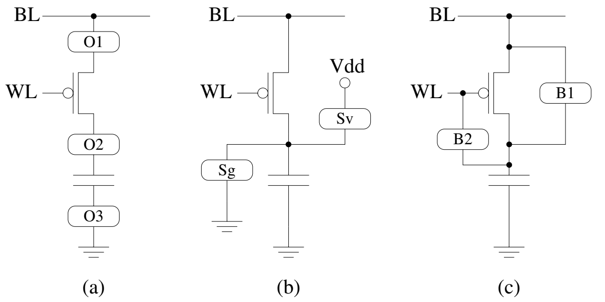


Figure 5. Simulated cell defects: (a) opens, (b) shorts and (c) bridges.

umn "Failing R" gives the value of the defect resistance above or below which the cell starts to fail. The column "Detection condition" gives a detection condition suitable to detect the faulty behavior resulting from the defect for any initial floating cell voltage V_c.

Table 1. Simulation results for defects shown in Figure 5.

Defect	Failing R	Detection condition
O1–3 (true)	$R \geq 200$ kΩ	$\Updownarrow(..., w1, w1, w0, r0, ...)$
O1–3 (comp.)	$R \geq 200$ kΩ	$\Updownarrow(..., w0, w0, w1, r1, ...)$
Sg (true)	$R \leq 1$ MΩ	$\Updownarrow(..., w1, r1, ...)$
Sg (comp.)	$R \leq 1$ MΩ	$\Updownarrow(..., w0, r0, ...)$
Sv (true)	$R \leq 400$ kΩ	$\Updownarrow(..., w0, r0, ...)$
Sv (comp.)	$R \leq 400$ kΩ	$\Updownarrow(..., w1, r1, ...)$
B1 (true)	$R \leq 200$ kΩ	$\Updownarrow(..., w0, r0, ...)$
B1 (comp.)	$R \leq 200$ kΩ	$\Updownarrow(..., w1, r1, ...)$
B2 (true)	$R \leq 200$ kΩ	$\Updownarrow(..., w0, r0, ...)$
B2 (comp.)	$R \leq 200$ kΩ	$\Updownarrow(..., w1, r1, ...)$

Each defect listed in the table has an entry (true) for defects in cells on the true bit line, and an entry (comp.) for defects in cells on the complementary bit line. The failing R value is the same for true and comp. entries. The detection conditions for the comp. entries have the same structure as their true counterparts, but with 1s and 0s exchanged.

The table shows that all defects start to fail in the resistance range 200 kΩ $\leq R \leq$ 1 MΩ. Open resistances, in particular, start to cause faults above a value of 200 kΩ, which is a relatively high value when compared to the open-channel resistance of 4.5 kΩ for the pass transistor. This indicates, for example, that the signal lines within cells are rather insensitive to process variations, and that strong opens are needed to cause a failure in the cell.

The table shows that, in order to detect any faulty behavior caused by cell *shorts* and *bridges*, detection conditions are needed with only 2 operations. The first operation of them is for initialization and the second is for sensitizing and detecting the fault. This, in turn, means that shorts and bridges cause *static* faults with $\#O$ = 1.

For faults caused by *opens*, a detection condition with four memory operations is needed. The first three write operations sensitize the fault and the fourth read operation detects the fault. This means that opens cause *dynamic* faults with $\#O$ = 3, which implies that the faults are not detected by most traditional tests, such as March C–, PMOVI, March LR and March LA.

5 Conclusions

In this paper, a new fault analysis approach, the approximate simulation, has been introduced. When compared with the conventional analysis, the approximate simulation approach have been shown to provide a speedup that increases exponentially with the number of operations analyzed. In addition, analysis methods have been given that use the approximate simulation to provide a deeper understanding of the faulty behavior. The approximate analysis approach is perfectly suited for analyzing the dynamic faulty behavior of memories since the analysis time is independent of the number of operations considered. Resistive defect injection and simulation of a DRAM model was used to validate the new analysis approach.

Acknowledgments

The authors wish to thank Jens Braun and Detlev Richter of Infineon Technologies, Munich, for their contributions, comments and positive feedback on this paper.

References

[Al-Ars99] Z. Al-Ars, *Analysis of the Space of Functional Fault Models and Its Application to Embedded DRAMs*, Technical Report no. 1-68340-28(1999)-07, CARDIT, Delft Univ. of Technology, Delft, The Netherlands, 1999.

[Al-Ars00] Z. Al-Ars and A.J. van de Goor, "Impact of Memory Cell Array Bridges on the Faulty Behavior in Embedded DRAMs," *in Proc. Asian Test Symp.*, 2000, pp. 282–289.

[Al-Ars01] Z. Al-Ars and A.J. van de Goor, "Static and Dynamic Behavior of Memory Cell Array Opens and Shorts in Embedded DRAMs," *in Proc. Design, Automation and Test in Europe*, 2001, pp. 496–503.

[Al-Ars02] Z. Al-Ars and A.J. van de Goor, "Modeling Techniques and Testing for Partial Faults in Memory Devices," to appear in Proc. Design, Automation and Test in Europe, Paris, 2002.

[vdGoor98] A.J. van de Goor, *Testing Semiconductor Memories, Theory and Practice*, ComTex Publishing, Gouda, The Netherlands, 1998, http://ce.et.tudelft.nl/~vdgoor/

[vdGoor00] A.J. van de Goor and Z. Al-Ars, "Functional Memory Faults: A Formal Notation and a Taxonomy," *in Proc. IEEE VLSI Test Symp.*, 2000, pp. 281–289.