DRAM Specific Approximation of the Faulty Behavior of Cell Defects

Zaid Al-Ars   Ad J. van de Goor
Section CE, 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: *To limit the exponential complexity required to analyze the dynamic faulty behavior of DRAMs, algorithms have been published to approximate the faulty behavior of DRAM cell defects. These algorithms, however, have limited practical application since they are based on generic memory operations (writes and reads) rather than the DRAM specific operations (activation, precharge, etc.). This paper extends the approximation algorithms by incorporating the DRAM specific operations, making them directly applicable in practice. In addition, based on the new extended method, the paper shows results of a fault analysis study of cell defects using electrical simulation.*

Key words: *memory specific fault analysis, approximating dynamic behavior, DRAM, memory testing*

1 Introduction

The ever increasing density and complexity of memory devices, and the gradual fragmentation of the memory market into a number of segments with different memory requirements, resulted in a number of special memory types, each with its own interfacing requirements and specific type of faulty behavior. As a result, research on the faulty behavior of memories is gradually turning away from generic methods to test memories in general toward memory specific methods which deal with the each type of memory separately.

It has been shown that it is possible to approximate the infinite dynamic faulty behavior of DRAM cell defects, using defect injection and electrical simulation of a memory model [Al-Ars02]. The method applies generic memory operations ($w1$, $w0$ and r) that are not DRAM specific. These operations are not sufficient to describe additional modes of memory operation (such as data refresh, fast page mode, etc.), which have a large impact on the possible faulty behavior of the memory [vdGoor00]. Work also has been published to account for DRAM operations in the context of fault description [Al-Ars01] and in the context of test generation [Vollrath00].

This paper extends the dynamic behavior approximation method to account for DRAM specific operations and operation sequences. This enables deriving detection conditions that are directly applicable to DRAMs, and enables DRAM specific test generation.

Section 2 identifies the DRAM specific operations and the allowed operation sequences. Then, Section 3 describes the method to approximate infinite dynamic faulty behavior of DRAMs. Section 4 extends the approximate method to account for DRAM specific operations, while Section 5 presents the results of the fault analysis performed to validate the extended method. Section 6 ends with the conclusions.

2 DRAM specific operations

DRAMs today have many modes of operation that aim primarily at increasing the performance by reducing the time needed to access stored information. Traditionally, memory functionality is described by simple r and w operations. However, DRAMs today can perform these two operations in many different modes, enabling more flexibility and/or more speed in manipulating the stored data. In order to describe the different DRAM modes of operation, five commands should be used that are more primitive than r and w. These commands are represented in Figure 1 and are described next:

1. Act: This is the *activate* command. When this command is issued, a word line (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 one of the sense amplifiers is moved to the I/O 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 I/O 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.

1081-7735/02 $17.00 © 2002 IEEE
IEEE COMPUTER SOCIETY

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

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

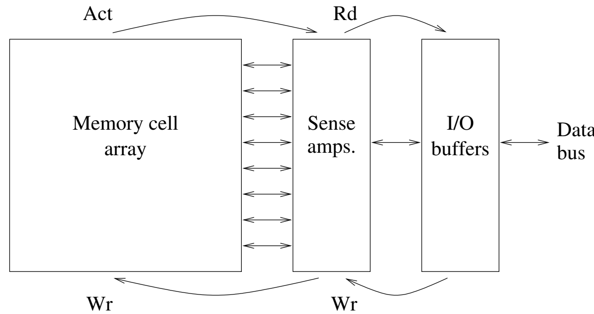

Figure 1. Functional model of a DRAM.

With the above five primitive commands, any DRAM operation can be described. A number of operation modes are described next using these five primitive commands.

Write operation: The write operation is traditionally denoted as wd_c, where d is the data to be written into cell c. Using the five primitive commands, the write operation is performed as Act$_{c_r}$ Wrd_{c_c} Pre, where c_r is the row address (or the WL address) of c and c_c is the column address (or bit line address) of c.

Read operation: The read operation is traditionally denoted as rd_c, where d is the expected data to be read from cell c. Using the five primitive operations, the read operation is performed as Act$_{c_r}$ Rdd_{c_c} Pre.

Refresh operation: The refresh operation is used to restore data into memory cells to prevent losing stored data by leakage. This operation cannot be represented using the traditional r and w operations. Using the five primitives, the refresh operation is performed as Act$_{c_r}$ Pre for all rows c_r in the memory.

Read modify write operation: This operation performs a read followed by a write on the same cell without the need to precharge the cell in between. This operation cannot be represented using the traditional operations. Using the five primitives, this operation is performed as Act$_{c_r}$ Rdx_{c_c} Wry_{c_c} Pre.

Fast page mode: In this mode, Rd and Wr operations are performed on any cell on a given activated WL (page) before precharging. This mode of operation greatly increases the performance of the memory. Using the five primitive operations, the fast page mode is performed as Act$_{c_r}$ O$d1_{c_{c1}}$ ... O$dn_{c_{cn}}$ Pre, where O is either Rd, Wr or Nop.

3 Approximating dynamic behavior

This section describes the method to approximate the infinite dynamic behavior for DRAM cell defects. Consider the defective DRAM cell shown in Figure 2, where an resistive open (R_{op}) reduces the ability to control and observe the voltage across the cell capacitor (V_c). The analysis takes a range of possible open resistances (1 kΩ $\leq R_{op} \leq$ 10 MΩ) and possible cell voltages (GND $\leq V_c \leq V_{dd}$) into consideration.

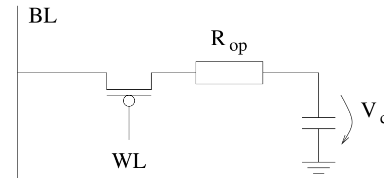

Figure 2. Electrical model of memory cell.

Next, a number of R_{op} values are selected for which the analysis is to be performed. Three different (V_c, R_{op}) result planes are generated, one for each memory operation ($w0$, $w1$, and r). These result planes describe the impact of successive $w0$, successive $w1$, and successive r operations on V_c, for a given value of R_{op}. Figure 3 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 3(a). To generate this figure, the floating cell voltage V_c is initialized to V_{dd} (because a $w0$ operation is performed) and then the operation sequence $1w0w0...w0$ is applied to the cell. 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, resulting in a number of curves. Each curve is indicated by an arrow pointing in the direction of the 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 $\Delta V_c \leq$ 0.24 V, since simulations show that this voltage is small enough. This selection of ΔV_c results in identifying two to four different $w0$ curves in the plane. Initially, an arbitrary small value for ΔV_c is selected, but can afterwards be reduced if it turns out that more than two $w0$ operations are needed to describe the faulty behavior. The mid-point 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 sense amplifier threshold voltage (V_{sa}) is shown in the figure as a dotted line. V_{sa} is the cell voltage above which the sense amplifier reads a 1, and below which the sense amplifier reads a 0.

1081-7735/02 $17.00 © 2002 IEEE

IEEE COMPUTER SOCIETY

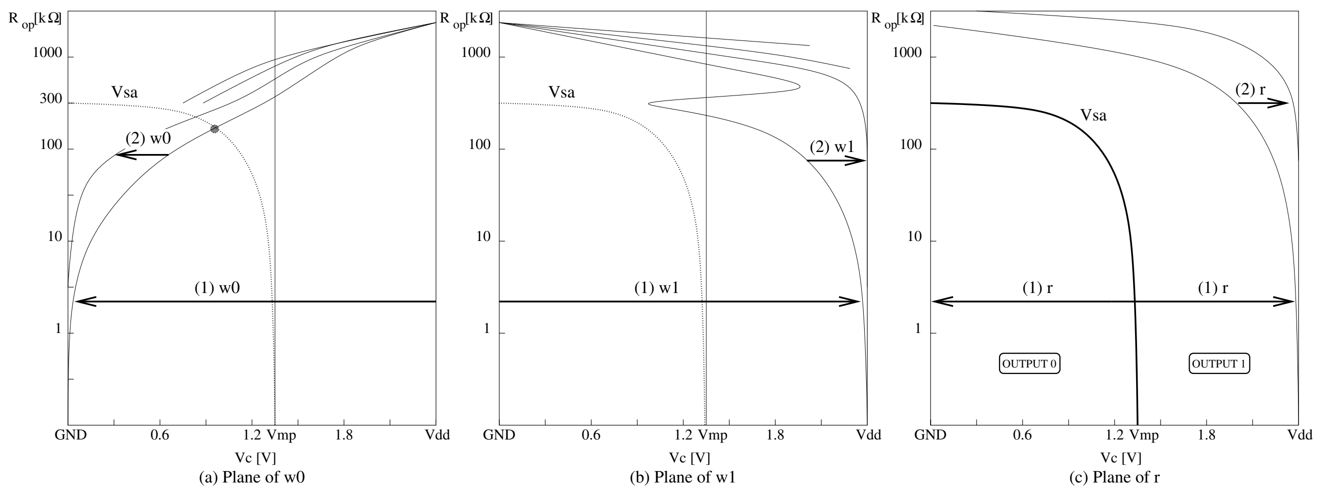

Figure 3. Result planes for the operations (a) $w0$, (b) $w1$, and (c) r.

Plane of $w1$: This result plane is shown in Figure 3(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 a gradual increase of V_c towards V_{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 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 in the curve $(1)w1$ of Figure 3(b) at about $R_{op} = 300$ kΩ. Remember that any memory operation starts with the sense amplifier sensing the voltage stored in the cell and amplifying it. Above R_{op} = 300 kΩ and for any stored cell voltage V_c, 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. V_{sa} is shown in the figure as a dotted line.

Plane of r: This result plane is shown in Figure 3(c). To generate this figure, first V_{sa} is established and indicated on the result plane (shown as a bold curve in the figure). As R_{op} increases, V_{sa} turns closer to GND which means that it gets easier to detect a 1 and more difficult to detect a 0.[1] Then the sequence $rrr...r$ is applied twice: first for V_c that is initially slightly lower than V_{sa} (0.12 V lower in this example), and a second time for V_c that is slightly higher than V_{sa} (0.12 V higher). 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 V_c to GND.

It is possible to use the result planes of Figure 3 to analyze a number of important aspects of the faulty behavior [Al-Ars02]. One such aspect relevant to this paper is the *border resistance (BR)*, which is the R_{op} value where the cell *starts* to cause faults on the output. For the faulty behavior shown in Figure 3, BR has a value of 200 kΩ, which is the value of R_{op} at the intersection between the $(1)w0$ curve and the V_{sa} curve (indicated as a dot in Figure 3(a)).

Another aspect relevant to this paper is generating a test that detects the faulty behavior of the defect. In the case of Figure 3, faults can be detected with $R_{op} \geq 200$ kΩ using the sequence $w1w1w0r0$. Note that the two $w1$ operations are necessary to charge V_c up fully to V_{dd} when R_{op} has a value close to BR. Performing one $w1$ instead of two, might charge V_c up to a voltage below V_{dd} which makes it easier for the subsequent $w0$ operation to write a 0. Instead of two $w1$ operations, one $w1$ with relaxed timing is also able to charge the cell up to V_{dd}, but this paper does not take timing into consideration, and all operations are assumed to have the same timing.

4 DRAM specific extension

In this section, the approximate analysis method is extended to account for the DRAM specific operations. It is discussed by applying it to the defect shown in Figure 2.

4.1 Selecting DRAM sequences

The approximate method presented in Section 3 shows that it is possible to use three sequences, called *basic sequences*, to approximate the dynamic faulty behavior of the memory. In order to extend the method to DRAM specific

[1] This is caused by the fact that the precharge cycle sets the bit line voltage to V_{dd}. Therefore, as R_{op} increases, a 0 stored in the cell fails to pull the bit line voltage down during a read operation, and the sense amplifier detects a 1 instead of a 0.

IEEE COMPUTER SOCIETY

Table 1. Basic operation sequences used to reconstruct any DRAM sequence.

#	C group	#	D group	#	C/D group	#	N group
1.	`0 Wr1 Wr1 Wr1 ... Wr1`	5.	`1 Wr0 Wr0 Wr0 ... Wr0`	9.	`Act Rd Rd ... Rd`	13.	`Pre Nop Nop ... Nop`
2.	`0 Wr1 Rd1 Rd1 ... Rd1`	6.	`1 Wr0 Rd0 Rd0 ... Rd0`	10.	`Act Nop Nop ... Nop`		
3.	`0 Wr1 Nop Nop ... Nop`	7.	`1 Wr0 Nop Nop ... Nop`	11.	`Act Act Act ... Act`		
4.	`0 Wr1 Pre Pre ... Pre`	8.	`1 Wr0 Pre Pre ... Pre`	12.	`Act Pre Pre ... Pre`		

operations, we need to find those (DRAM specific) basic sequences that make such an approximation possible.

In order to do this, the chosen set of DRAM specific basic sequences has to answer the following question: Given an arbitrary point in the (R_{op}, V_c) plane, how much approximately is the voltage change (ΔV_c) resulting from performing a given DRAM operation? This problem is depicted in Figure 4, where Op stands for any DRAM operation, V_{c1} stands for V_c before Op is applied, and V_{c2} stands for V_c after Op is applied.

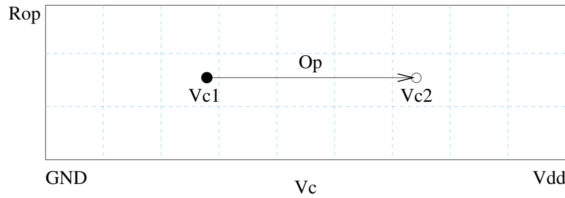

Figure 4. Effect of Op on the V_c in the DRAM result plane.

Table 1 lists 13 DRAM specific basic sequences able to approximate ΔV_c for any Op. The basic sequences are classified into four different groups: charge up group (C group), discharge group (D group), charge/discharge group (C/D group), and no charge group (N group). The name in each group refers to the direction of expected voltage change within the memory cell during the application of the sequence. In the following, we show how these basic sequences can be used to approximate the impact of any Op on V_c.

Act: As discussed in Section 2, activation performs an internal read operation by accessing a row of cells and sensing the data they contain. Starting with V_c above V_{sa}, Act increases V_c, while with V_c below V_{sa}, Act reduces V_c. Therefore to identify the impact of Act on V_c, Sequence 11 in Table 1 is needed and should be performed twice: starting with V_c above V_{mp} and with V_c below V_{mp}.

Pre: Precharging closes the activated row of cells and sets the voltage of bit lines and data lines to their precharge levels. During this operation, the change in V_c is decided by the previously performed operation, which might be Act, Wr1, Wr0, Rd or Nop. Remember that Rd only performs an external read operation, which means that it has little impact on internal behavior. The same is true for Nop, which simply represents a wait state and does not influence subsequent Pre operations. It is important, however, to identify the impact of Pre on V_c with a previously performed Wr1 (Sequence 4), with a previously performed Wr0 (Sequence 8), and with a previously performed Act (Sequence 12).

Rd: Reading performs an external read operation from the sense amplifiers to the I/O buffers. The impact of Rd on V_c depends on the previously performed operations. Therefore, Sequence 2 identifies the impact for a previously performed Wr1, Sequence 6 for performed Wr0, and Sequence 9 for performed Act.

Wr: The impact of writing operations on V_c does not depend on the previously performed operation, but on the data being written. Therefore, it is important to inspect the behavior of Wr1 operations (Sequence 1) and the behavior of Wr0 operations (Sequence 5). When inspecting Wr1, the cell should be initialized to 0 since this represents the worst case for writing 1. When inspecting Wr0, the cell should be initialized to 1 since this represents the worst case for writing 0.

Nop: The no operation represents wait states where the memory is supposed not to change its state. During Nop, however, the value of V_c does change and it, therefore, needs to be analyzed. The impact of Nop on V_c depends on the previously performed operations. Sequence 3 evaluates the impact of Nop after Wr1, Sequence 7 after Wr0, Sequence 10 after Act, and Sequence 13 after Pre.

4.2 Sequence application

To keep the analysis in this paper clear and simple, we assume that faults can only be sensitized using the operations performed between Act and Pre operations, since the cell is only accessed between Act and Pre. Therefore, we can exclude from our analysis Sequences 4, 8 and 12 in Table 1 since they inspect the impact of Pre operations, exclude Sequence 11 since it inspects the impact of Act operations, and exclude Sequence 13 since it inspects the impact of Nop operations performed after Pre and before Act. This leaves Sequences 1, 2 and 3 of the C group, Sequences 5, 6 and 7 of the D group, and Sequences 9 and 10 of the C/D group.

IEEE COMPUTER SOCIETY

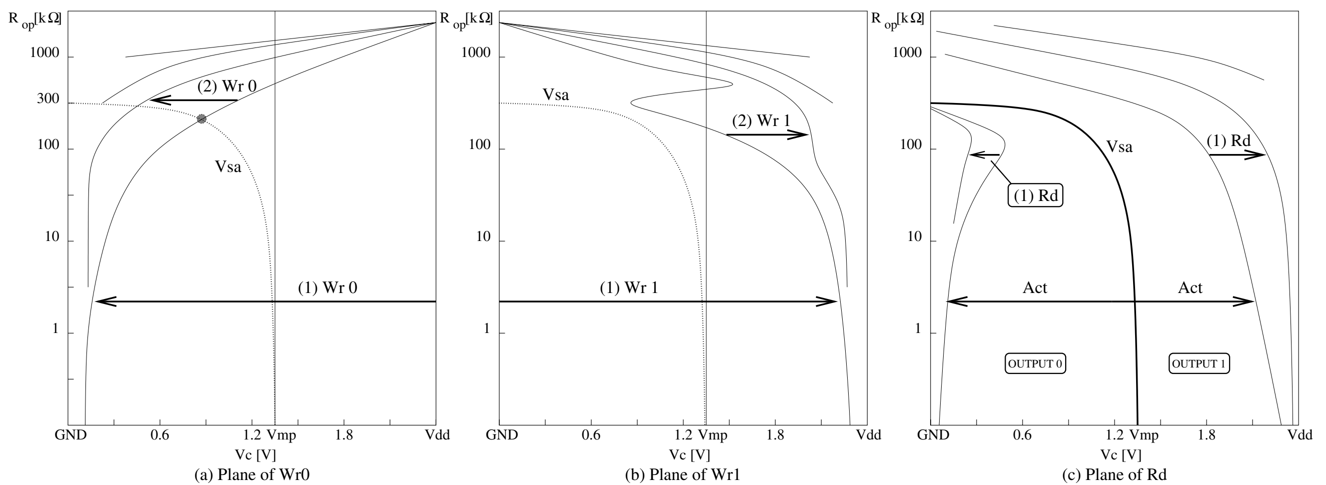

Figure 5. Result planes for the DRAM specific sequences (a) Wr0, (b) Wr1, and (c) Rd.

Fortunately, we do not need to generate all 8 result planes for these 8 sequences. Simulations show that, for most defects, the sequences remaining in each group have similar impact on the faulty behavior, which means that the result planes they generate are similar, and therefore only one sequence in each group needs to be simulated as a representative sequence. Simulations show that this is true for all defects within DRAM cells treated in this paper.

We select the following sequences as representatives for the impact on faulty behavior:

- `0   Wr1 Wr1 ... Wr1` (Sequence 1 in Table 1)
- `1   Wr0 Wr0 ... Wr0` (Sequence 5)
- `Act Rd  Rd  ... Rd` (Sequence 9)

Figure 5 shows the result plane for each of these sequences. The results are similar to those shown in Figure 3 performed using the simple sequences (w and r). Both figures show, for example, that the memory starts to fail at an R_{op} of about 200 kΩ. This indicates that the possible fault coverage did not change as a result of using the DRAM specific operations instead of the generic memory operations. On the other hand, the detection condition needed to detect the faulty behavior changes as a result of analyzing the DRAM operations. The faulty behavior for $R_{op} \geq 200$ kΩ and higher can be detected using the detection condition $\Updownarrow$(..., Wr1, Wr1, Wr0, Pre, Act, Rd0, ...).

To compare this detection condition with the one derived for the generic memory operations, we construct a test out of it as follows:

$$\begin{array}{ll} & \Updownarrow(\text{Act, Wr1, Wr1, Wr0, Pre, Act, Rd0, Pre}) \\ \text{replaces} & \Updownarrow(\quad w1,\ w1,\ w0 \qquad\qquad r0) \end{array}$$

Each w operation should start with an Act and end with Pre. Assuming that each of the five primitive DRAM operations takes one clock cycle, then using DRAM specific operations reduces test time from $4 \times 3 = 12$ to 8 operations, which is a reduction by $4/12 = 33\%$.

5 Fault analysis results

This section presents the results of a fault analysis performed on a number of DRAM cell defects to verify the new extended DRAM method. The simulation approach is presented first, and thereafter the application results are discussed.

5.1 Simulation approach

The simulation model used in the analysis has one folded bit line pair with 2 cells on the true bit line (BT) and 2 cells on the complementary bit line (BC). The model uses V_{dd} as the BL precharge voltage, and sensing is done using 2 reference cells. In addition, the model has one sense amplifier, one write driver and one data output buffer to inspect the data forwarded to the output. Simulation has been performed using an HSpice based simulation tool.

Figure 6(a) shows the 3 simulated cell opens: open on top of the cell (Ot), open in the middle (Om), and open at the bottom (Ob). Figure 6(b) shows the simulated 2 cell shorts: short to V_{dd} (Sv), and short to GND (Sg). Figure 6(c) shows the 2 simulated cell bridges: bridge to word lines (Bw), and bridge to bit lines (Bb). The simulated defect resistance values are taken in the range (1 kΩ $\leq R \leq$ 10 MΩ) on a logarithmic scale.

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

IEEE COMPUTER SOCIETY

Table 2. Simulation results for defects shown in Figure 6, using generic and DRAM specific operations.

Defect	Failing R	Detection condition	Failing R	DRAM specific detection condition	Time reduction
Ot,m,b (true)	$R \geq 200$ kΩ	$\Updownarrow(..., w1, w1, w0, r0, ...)$	$R \geq 200$ kΩ	⇕(..., Wr1, Wr1, Wr0, Pre, Act, Rd0, ...)	33%
Ot,m,b (comp.)	$R \geq 200$ kΩ	$\Updownarrow(..., w0, w0, w1, r1, ...)$	$R \geq 200$ kΩ	⇕(..., Wr1, Wr1, Wr0, Pre, Act, Rd0, ...)	33%
Sg (true)	$R \leq 1$ MΩ	$\Updownarrow(..., w1, r1, ...)$	$R \leq 1$ MΩ	⇕(..., Wr1, Pre, Act, Rd1, ...)	0%
Sg (comp.)	$R \leq 1$ MΩ	$\Updownarrow(..., w0, r0, ...)$	$R \leq 1$ MΩ	⇕(..., Wr0, Pre, Act, Rd0, ...)	0%
Sv (true)	$R \leq 400$ kΩ	$\Updownarrow(..., w0, r0, ...)$	$R \leq 400$ kΩ	⇕(..., Wr0, Pre, Act, Rd0, ...)	0%
Sv (comp.)	$R \leq 400$ kΩ	$\Updownarrow(..., w1, r1, ...)$	$R \leq 400$ kΩ	⇕(..., Wr1, Pre, Act, Rd1, ...)	0%
Bb (true)	$R \leq 200$ kΩ	$\Updownarrow(..., w0, r0, ...)$	$R \leq 200$ kΩ	⇕(..., Wr0, Pre, Act, Rd0, ...)	0%
Bb (comp.)	$R \leq 200$ kΩ	$\Updownarrow(..., w1, r1, ...)$	$R \leq 200$ kΩ	⇕(..., Wr1, Pre, Act, Rd1, ...)	0%
Bw (true)	$R \leq 200$ kΩ	$\Updownarrow(..., w0, r0, ...)$	$R \leq 200$ kΩ	⇕(..., Wr0, Act, Pre, Rd0, ...)	0%
Bw (comp.)	$R \leq 200$ kΩ	$\Updownarrow(..., w1, r1, ...)$	$R \leq 200$ kΩ	⇕(..., Wr1, Pre, Act, Rd1, ...)	0%

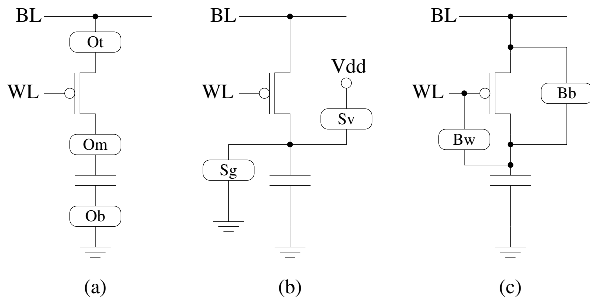

Figure 6. Simulated cell defects: (a) opens, (b) shorts and (c) bridges.

5.2 Simulation results

Table 2 gives a summary of the simulation results. The table lists the simulated defects in the first column, then the BR and detection condition using the generic memory operations, followed by the BR and detection condition using the DRAM specific operations. In addition, the time reduction in the detection condition as a result of using the DRAM operations is given.

Each defect listed in the table has an entry (true) for defects in cells on the true bit line (BT), and an entry (comp.) for defects in cells on the complementary bit line (BC). The value of BR 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 it is possible to use the approximation method to generate DRAM specific detection conditions that better suit the practical needs of DRAMs, and that are directly transformable into DRAM specific tests. The table also shows that the value of the two BRs is the same for all defects. This is a natural result of the definition of BR, since it represents the R_{op} value where faults start to take place for any possible sequence of operations. Therefore, the value of BR is more related to the defect itself than it is related to the type of sequence that detects it (as long as stresses are kept the same).

Only opens within the cell (Ot, Om, Ob) result in a reduction in the test time when DRAM operations are considered. The reason is that only very simple sequences are needed to detect the faulty behavior of other defects (a simple write read sequence). If a defect requires more complex sequences (like those needed for the opens) then time reduction in the test sequence can be observed. However, the advantage of considering DRAM specific operations does not lay in reducing the required test time, but in the ability to analyze and represent DRAM faults using DRAM specific operation sequences.

6 Conclusions

In this paper we have shown how to perform fault analysis in a DRAM specific way using DRAM specific operations. An algorithm to approximate the infinite dynamic behavior of the DRAM is extended to take DRAM operation into consideration. In addition, the algorithm has been applied to a number of DRAM cell defects and the results of the analysis are discussed.

References

[Al-Ars01] Z. Al-Ars *et al.*, "A Memory Specific Notation for Fault Modeling," *in Proc. Asian Test Symp.*, 2001, pp. 43–48.

[Al-Ars02] 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.

[vdGoor00] A.J. van de Goor and A. Paalvast, "Industrial Evaluation of DRAM SIMM Tests," *in Proc. IEEE Int'l Test Conf.*, 2000, pp. 426–435.

[Vollrath00] J. Vollrath, "Tutorial: Synchronous Dynamic Memory Test Construction, A Field Approach," *in Proc. IEEE Int'l Workshop Memory Technology, Design and Testing*, 2000.

1081-7735/02 $17.00 © 2002 IEEE
IEEE COMPUTER SOCIETY