A New Test Paradigm for Semiconductor Memories in the Nano-Era

Said Hamdioui Venkataraman Krishnaswami Ijeoma Sandra Irobi Zaid Al-Ars
Computer Engineering Laboratory, Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
{S.Hamdioui, I.S.Irobi, Z.Alars}@tudelft.nl; V.Krishnaswami-1@student.tudelft.nl

Abstract—Due to rapid and continuous technology scaling, faults in semiconductor memories (and ICs in general) are becoming pervasive and weak rather than strong; a weak fault is a fault that escape the test program (because it does not cause an error/ system failure). However, multiple weak faults may cause an error during the application. Components with weak faults which fail at board and system level are sent to suppliers, but only to have them returned back as No Trouble Found (NTF). This is because the conventional memory test approach assumes the presence of a single defect at a time causing a strong fault (hence an error), and is therefore unable to deal with weak faults. This paper presents a new memory test approach able to detect weak faults; it is based on assuming the presence of multiple weak faults at a time in a memory system rather than a single strong fault at a time. Being able to detect weak faults reduces the number of escapes, hence also the number of NTFs. The experimental analysis done using SPICE simulation for a case of study show that when assuming two simultaneous weak faults, the missing (defect) coverage can be reduced with 10% as compared with the conventional approach.

Index Terms—Memory Test, fault models, strong faults, weak faults, defect injection and circuit simulation

I. INTRODUCTION

Progressive technology scaling, as tracked by the International Technology Roadmap for Semiconductors (ITRS) and encapsulated by Moore’s law [1], has driven the phenomenal success of the semiconductor industry. Silicon technology has now entered the nano-era and the 10nm transistors are expected to be in production by 2018. This will allow for the integration of a wider variety of functions. However, it is widely recognized that variability in device characteristics and its impact on the overall quality and reliability of the system represent major challenges to scaling and integration for present and future nanotechnology generations [2], [3]. What is more, newly emerging complex failure mechanisms in the nano-era (which are not understood yet), are causing the fault mode of the chips to be dominated by transient, intermittent and weak faults rather than hard and permanent faults; hence causing more reliability problems than quality problems [4], [5], [6], [7]. Many companies are reporting not being able to explain all electronic failures with the existing approaches [8], [9], [10], [11]. For instance, AUDI reported from all electronic failures, 35% can be mapped using the existing approaches into well defined semiconductor failures, while 41% cannot be understood (NTF: No Trouble Found) [9]. This shift in failure mechanisms is therefore seriously impacting the quality and reliability, which means that the design of future systems fabricated using nanotechnology is a major challenge. In turn, this will demand revolutionary changes in how future systems are designed and tested to meet the increasing quality requirements on such systems.

Nowadays, embedded memories represent the great majority of embedded electronics in Systems on Chip (SoC). It is very common to find SoCs with hundreds of memories representing more than 50% of the overall chip area. According to the ITRS, today’s SoCs are moving from logic-dominant to memory-dominant chips in order to deal with the requirements of todays and future applications. Consequently, embedded memory test challenges will significantly impact the overall testability of SoC. Solving such challenges for memories will substantially contribute to the resolution of electronic system test problems in the future; hence, supporting the continuation of the semiconductor technology revolution and the manufacturability of future highly complex systems (‘giga-scale’) and highly integrated technologies (‘nano-scale’).

The purpose of this paper is the development of a new foundation which can be used to systematically develop new fault models and test algorithms for failure mechanisms of memory systems in the nano-era. The new foundation is able to deal with complex faulty behaviors of memory systems, increase the defect coverage and enable the reduction of NTFs. It is based on simultaneously considering the presence of weak faults in different parts of the memory system (e.g., memory array and address decoder); this significantly impact the test detection and test development approach. The approach is validated using a SPICE simulation. The simulation results show that the missing defect coverage can be reduced with up to 10% for the considered case-of-study.

The paper is organized as follows. Section 2 presents an brief overview of the state-of-the art of embedded memory testing. Section 3 gives the notation used to describe memory fault models and test algorithms as these will be used in the rest of the paper. Section 4 discusses the traditional memory test approach. Section 5 presents the new memory test approach. Section 6 gives the simulation results to validate the approaches. Section 7 concludes the paper.

II. STATE-OF-THE ART IN MEMORY TESTING

Tests for semiconductor memories have undergone a long development process. Before 1980, tests for a given fault cov-
eral (FC) were time consuming (FC is defined as the number of detected faults divided by the number of total faults) [12]. Such tests can be termed as the ‘ad hoc’ tests because they were not based on fault models and proofs. To reduce the test time and improve the FC, test development has been focused on possible faulty behaviors in the memory at the functional level. For that reason, functional fault models, which are abstract fault models, were introduced during the early 1980s. After that, ‘March’ tests have become dominant because their test times are in linear proportion with the size of the memory, and their FC can be proven mathematically [13], [14], [15]. In the late 1990s, industrial results indicated that many tests detect faults which could not be explained by the existing fault models at the time [16], [17], [18] mainly because of the used technology node. This led to the introduction of new concepts, new fault models and test schemes [19]-[22]. Today, as the silicon industry moves towards the end of the CMOS technology roadmap, controlling the fabrication of scaled memory devices is becoming a major challenge. Device-parameter variations (e.g. threshold voltage), high defect density as well as new failure mechanisms in the nano-era are expected to be significantly larger in the future [3], [5], [23], [24], and this will create major challenges in designing and testing memories in nanotechnology. Conventional fault models and test approaches are inadequate to realize the required product quality [5], [6], [10], [11], [25], [9], [26]. In the absence of new theories capable of modeling their failures mechanism and developing appropriate test solutions, the production of future electronics systems will become infeasible.

Given the state-of-the-art in fault modeling and test generation for embedded memory, one can conclude there are, as yet, no fault models to describe the above failure mechanisms. The models have to consider not only the presence of a single defect at a time (as it has been the case so far), but it also has to consider the presence of multiple weak-faults/defects simultaneously (this is particularly important in the nano-era). A weak fault is not able to fully sensitize a fault, but it partially sensizes a fault; e.g., due to a defect that creates a small disturbance of the voltage at the true node of the cell. However, a fault can be fully sensitized (i.e., becomes strong) when two (or more) weak faults are sensitized simultaneously in the memory systems since their fault effects can be additive.

This paper will advance the knowledge in the field of memory fault modeling and test generation by introducing a new theoretical foundation being able to deal with the complex faulty behavior of embedded memories in the nano-era. A systematic approach will be developed in order to accurately model the memory faulty behavior, and develop appropriate test algorithms and solutions. The approach fundamentally differs from the conventional approach in two aspects: (a) it considers not only strong faults/defects, but also weak faults and parametric deviations as they may impact the memory reliability as well, (b) it considers not only single fault/defect at a time, but also multiple-faults/defects simultaneously in different parts of the memory system (e.g., memory array and address decoder). This, in turn, will allow the explanation of some of today’s not understood faults, and facilitate the understanding and modeling of future electronic failures in 32nm technology and beyond. The approach will increase the defect coverage, reduce the number of escapes and enable the reduction of NTFs.

III. FAULT MODELS AND ALGORITHMS NOTATION

This section describes the fault model and algorithms notation that will be used in this paper.

In order to specify a certain memory fault, one has to represent it in the form of a fault primitive (FP) [19], denoted as $<S/F/R>$. $S$ describes the operation sequence that sensitizes the fault, $F$ describes the logic level in the faulty cell ($F \in \{0,1\}$), and $R$ describes the logic output level of a read operation ($R \in \{0,1,-\}$). $R$ has a value of 0 or 1 when the fault is sensitized by a read operation, while the ‘-’ is used when a write operation sensitizes the fault. For example, in the FP $<0w1/0/->$, which is the up-transition fault (TF1), $S=0w1$ means that a w1 operation is written to a cell initialized to 0. The fault effect $F=0$ indicates that after performing $w1$, the cell remains in state 0. The output of the read operation ($R=-$) indicates there is no expected output for the memory.

The algorithms in this paper are described with an extended notation for march algorithms. March algorithms are the most common algorithms used for testing memories [15]. A march test consists of a finite sequence of march elements. A March Element (ME) is a finite sequence of operations applied to every cell in the memory before proceeding to the next cell. The way one proceeds to the next cell is determined by the Address Orders (AOs) which can be an increasing address order (e.g., increasing AO from the cell 0 to the cell $n-1$), denoted by ↑ symbol, or a decreasing AO, denoted by ↓ symbol, and which is the exact inverse of the ↑ AO. When the AO is irrelevant, the symbol ⇑ (i.e., ↑ or ↓) will be used. A complete march test is delimited by the ‘[...’ bracket pair; while a march element is delimited by the ‘(…)’ bracket pair. The march elements are separated by semicolons, and the operations within a march element are separated by commas. An example of a march algorithm is MATS+ [29], defined as: $\{\uparrow(w0); \downarrow(r0, w1); \downarrow(r1, w0)\}$. MATS+ consists of three MEs, which are separated by the ‘;’ symbol. The ME ‘$\uparrow(r0, w1)$’ specifies the ↑ AO, while to each address a read operation with expected value ‘0’ will be applied, after which a ‘1’ will be written.

IV. TRADITIONAL MEMORY TEST APPROACH

Figure 1 shows a simplified block diagram of a semiconductor memory. The figure distinguishes three main blocks which are usually considered for fault modeling and test purposes: memory cell array, the address decoders (e.g., address latches, row decoders, column decoders) and the peripheral circuits (e.g., write drivers, sense amplifiers).
Traditionally fault modeling is done by injecting a single defect at a time -such as a resistor- in the array, the address decoder or in the peripheral circuit. Electrical simulation (e.g., SPICE) is then performed and the faulty behavior is analyzed using either static or dynamic analysis [27], [19], [20]. Depending on how severe the defect is (e.g., how big is the value of the resistor representing the defect), the fault may or may not be sensitized. If the fault is sensitized and an error is caused, then it is mapped into a functional fault model. An appropriate detection condition is then developed and compiled thereafter in a test algorithm.

Let's consider the case where a single defect will be injected in the memory cell array. Assume that the connection between the cross-coupled invertors of the memory cell array suffer from a resistive open defect. An appropriate memory model has been built for the simulation; it includes a 4x4 memory array, sense amplifiers, write drivers, pre-charge circuits, address decoders, etc. The simulation has been done for the sequence $S = w0, r0, w1, r1$. Figure 2 shows the voltage at the true node of the memory cell for three cases: (a) defect free case (top graph), (b) high value of the defect (middle graph), and (c) small value of the defect (bottom graph). The middle graph shows that the $w1$ operation fails and that the state of the cell will remain in 0, causing "Transition Fault" ($<0\rightarrow1/0\rightarrow-\rightarrow$). Inspection of the bit line voltages and the output of the sense amplifier (not included in the figure) in this case reveals that the $r1$ operation returns 0 instead of an expected 1. Hence, the sequence $S = w0, r0, w1, r1$ will detect the fault. In case the defect is not high enough, all the operations will pass and no fault will be observed at the output of the sense amplifier; hence no error. Nevertheless, careful inspection of the memory cell (see bottom figure) shows that the transition from 0 to 1 does not pass smoothly as it is the case for defect-free (see also the top graph). Hence, even externally it seems that there is no trouble, in reality the cell suffers from small disturbance which is not strong enough to cause an error; we call this fault a weak fault; in this case a weak transition fault.

Figure 3 illustrates the difference between the Fail (strong fault) and Pass (weak faults) regions depending on the value of the defect as we externally experience it. If the defect is beyond a certain value, say $R_{\text{critical}}$, then a strong fault will be sensitized and detected. However, below this value, the fault will be not detected. Because when performing memory fault modeling and test generation we assume the presence of a single fault at a time, a defect below say $R_{\text{critical}}$ (causing a weak fault) will never be detected. If we now assume the presence of another weak fault in another part of the memory cell (e.g. address decoder), then the fault effects of the two faults may be additive and sensitize an error. That is something that may occur during the application and cause the memory system to fail. A manufacturing test algorithm can detect such faults only if it combines the detection conditions of the two (or more) weak faults; that is the missing link in the traditional approach.

V. NEW MEMORY TEST APPROACH

As already mentioned, technology scaling is causing more complex failure mechanisms; having small disturbances in different parts of a memory system is a reality [10], [25], [26]. Therefore, taking the fault effects of such disturbances together into consideration while developing fault models and designing test algorithms is required not only to increase the outgoing product quality and reduce the number of escapes, but also to contribute to the reduction of NTFs as well. In the rest of this section, the new approach will be explained and illustrated.

As already mentioned, the basic idea is to consider the combined fault effects of multiple weak faults. In our case, we will use the reduced memory system consisting of three main blocks (see Figure 1): memory cell array, address decoders and peripheral circuits. We further assume that each of the three blocks may suffer from a weak fault. In case of a single-defect at a time approach (i.e. traditional approach), the faulty behavior of the memory can be graphically illustrated in a similar way as shown in Figure 3. If now, we consider the presence of two defects at a time, say defect R1 in the memory cell array...
and defect R2 in the address decoders, then the defect/fault coverage can be illustrated in a two dimensional graph as shown in Figure 4(a). The hashed areas in the graph represent the coverage that can be achieved based on the traditional approach. Obviously, this coverage will be also realized with the approach based on two defects at a time. Moreover, the new approach is able to improve the defect/fault coverage, which is represented by the region marked with '*' in the graph. Therefore, some weak faults that may escape the traditional test approach will be detected when assuming two defects at a time. It is worth noting that the way test algorithms are developed has to be adapted in such way that the sensitization conditions of both faults have to be considered simultaneously; i.e., the sensitization conditions have to combined into a single condition. For example, assume that R1 in the memory cell array causes a weak transition fault (see Figure 2) and R2 in the address decoder causes an activation delay (ActD) fault in the row decoder [30], [31]. The sensitization condition of TF requires the application of the transition w1 operation to a memory cell, while ActD fault requires special address transitions (such as Address Complement) and the use of fast row address direction. Combining these two condition will result in a transition write operations through fast row using e.g., address complement.

The defect/fault coverage can even be improved if more than two defects at a time are considered. Figure 4(b) illustrates the additional coverage that can be realized when assuming three simultaneous defects: defect R1 in the memory cell array, defect R2 in the address decoders and defect R3 in the peripheral circuit. As the figure shows, the coverage will be further improved as compared with Figure 4(a); this improvement is given by the regions marked with '+' in the figure. Note that the regions marked with '*' indicate the improvement when considering two simultaneous defects as compared with the traditional approach.

VI. SIMULATION RESULTS

An appropriate memory simulation model has been used; it consists of a 4x4 cell array, address decoders and each column has its own set of peripheral circuits such as sense amplifiers, write drivers, pre-charge circuits, etc. (see Figure 1). The used transistor parameters are as described for 45nm PTM models.

As a case of study, we considered the following three defects (see Figure 1):
- Defect R1: The connection between the cross-coupled inverters of the memory cell array suffer from a resistive open defect.
- Defect R2: A resistive open in one of the address line inputs of an NAND gate of the row decoder.
- Defect R3: A resistive open in the pull-up transistor of the inverter in the write driver circuit.

In the rest of this section the simulation results and analysis will be shown first for the traditional approach (i.e. assuming a single defect at a time), and thereafter for the new approach by assuming two or three weak faults simultaneously. We will also show how test algorithms can be developed in a systematic way in order to improve the defect/fault coverage.

A. Single defect approach

Each of the defects R1, R2 and R3 has been simulated; again only a single defect at a time is simulated. The fault behavior of the memory is analyzed and compiled into fault models; the results are summarized in Table I. The first column gives the defect, the second column the critical resistance \( R_c \) of each defect, and the third column the fault model observed. Note that for defects with resistance below \( R_c \), no error was observed, instead a weak fault was sensitized; see Figure 3. Hence, the fault will be not detected with tests used for strong faults. The observed strong faults consist of:
- Defect R1: A write 1 fails to change the state of the cell resulting into TF=\(< 0w1/0/− >\).

<table>
<thead>
<tr>
<th>DF</th>
<th>( R_c ) (Ω)</th>
<th>Fault model</th>
<th>Detection condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>489K</td>
<td>TF</td>
<td>A transition w1 followed by a read</td>
</tr>
<tr>
<td>R2</td>
<td>55K</td>
<td>ActD</td>
<td>Two back-to-back operations with opposite data values and using special address transitions with fast row</td>
</tr>
<tr>
<td>R3</td>
<td>91K</td>
<td>SWDF</td>
<td>Write-after-write back-to-back operations using opposite data values and fast row.</td>
</tr>
</tbody>
</table>
• Defect R2: The activation of the addressed word line is delayed due to the open defect; this results into an Activation Delay (ActD) fault [30], [31].
• Defect R3: The write driver becomes too slow due to the open defect in the driver; therefore, the differential voltage on the bit lines during the write operation is reduced, causing the cell not to be written [32], [14]. This fault is called Slow Write Driver Fault (SWDF).

The last column in the table lists the requirements a test has to satisfy in order to detect the corresponding fault. E.g., the ActD requires two back-to-back operations, say opx and opx where \( op \in \{0, 1\} \) and \( x \in \{0, 1\} \) have to be performed using special address transitions (such as address complement or hamming) with fast row [30], [31]; and the SWDF requires the application of two back-to-back write operations with opposite data values with fast row. Table II lists examples of minimal tests for each fault of Table I. Test for TF can use a linear address sequence (i.e., \( 1, 2, 3, ..., N \) where \( N \) is the number of addresses) with either fast column or fast row; the test for ActD fault has to use address complement sequence (denoted with \( AC \)) in the test) with fast row, while the test for SWDF can use the linear addressing with fast row.

The tests were simulated using our memory SPICE model for the three defects with values larger than their respective \( R_e \) in order to evaluate their coverage. The results are summarized in Table III; a ‘+’ in the table means that the test detects the corresponding fault; e.g., Test for ActD detects both TF and ActD but not SWDF. It is worth noting that the development of each of the tests is based on compiling a single detection condition into a test algorithm and that the detected faults are strong ones.

**B. Two weak faults simulation**

Next we will assume the presence of two faults at a time; the two defects R1 and R2 are injected in the memory system for the simulation where the values of the defects are smaller than their critical values \( R_e \) (R1=420KΩ and R2= 52KΩ). R1 and R2 can each cause a weak fault. The simulation is performed for three sequences (see also Table II): (a) Test TF, (b) Test ActD, and (c) new sequence generated by combining the detection conditions of TF and ActD faults (see Table I); this test is 'Test TFActD' with the following description: \{\( \hat{\cup}(wd) \), \( AC \hat{\cup}(wd) \), \( AC \hat{\cup}(rd) \), \( AC \hat{\cup}(wd) \), \( AC \hat{\cup}(rd) \)\}

where \( d \) denotes the checkerboard data-background [15]; the test is applied in fast row address direction. Note that this test satisfies the detection conditions of both TF and ActD fault.

Figure 5 gives the voltage at the true node of the memory cell suffering from the defect R1 (i.e., cell #8 in the 4x4 array under consideration) for the three simulated tests. The top graph in the figure shows the result for Test TF; it clearly shows the write operations pass correctly even in the presence of the defects. Inspection of the bit lines voltage and the output of the sense amplifier (not included in the figure) reveals that the read operation returns the correct value, hence the presence of two weak faults cannot be detected. The middle graph shows the results for Test ActD; also here the operations pass correctly and no fault has been externally observed. The bottom graph shows the results for Test SWDF; the cell fail to undergo an up write transition, hence the fault is sensitized. In addition, externally the fault was detected by read operation producing 0 instead of 1. Note that for this test checkerboard data-background \( d \) is used, and that \( d=0 \) for cell #8 in the 4x4 memory array under consideration.

The simulation has been performed for different values of R1 and R2 in order to identify the pass and fail regions. The results are plotted in Figure 6(a). An estimation of the realized defect coverage improvement can be done by calculating the size of the area marked with '×' and dividing it by the area defined by \( R_1 \) and \( R_2 \); this results in a reduction of 10% in the missing defect coverage.

Similar approach and simulation have been performed while considering simultaneously weak R2 and weak R3. The results show a reduction of 7% in the missing defect coverage can be realized. On the other hand, considering R1 and R3 simultaneously did not realize any defect coverage improvement as the detection condition for R3 also covers that of R1.

**C. Three weak fault simulation**

In this case three defects R1, R2 and R3 have been injected in different parts of the memory; the values of defects are smaller than their critical values (causing only weak faults). Thereafter, six test sequences have been simulated:

(a) The three tests targeting single-cell defect at a time as discussed in Section VI-A and given in Table II.
The two tests targeting two simultaneous weak faults such as Test TF&ACID discussed in Section VI-B.

A new test generated by combining all the three detection conditions into a single condition and compiling it into a single test.

The simulation results show that the sequence under (c) outperforms (in terms of defect coverage) all the sequences under (a) and under (b). Figure 6(b) shows the relationship between the defect/fault coverage for the different approaches. The more simultaneous weak faults considered when developing a test algorithm, the higher the realized coverage.

VII. CONCLUSIONS

In this paper, a new memory test paradigm has been presented. A new systematic approach being able to deal with complex faulty behavior of embedded memories in the nanoeera is presented. It is based on assuming the presence of two or more weak faults at a time in the memory system; such weak faults may have an additive fault effect and therefore cause an error. Detection of such faults requires deep understanding of each of the weak faults, development of detection conditions for each fault, and thereafter combining all the detection conditions into a single one being able to sensitize all the weak faults at once. The SPICE simulation done on memory model based on 45nm PTM transistor parameters reveals that by just considering two weak faults at a time, the missing defect coverage by the traditional approach can be reduced by 10%. The more combined weak faults, the higher the defect coverage.

REFERENCES