

Future Challenges in Memory Testing

Said Hamdioui

Georgi Gaydadjiev

Delft University of Technology, Faculty of EEMCS

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

E-mail: {S.Hamdioui, G.N.Gaydadjiev}@ewi.tudelft.nl

Tel: +31- 15278 36 43; Fax: +31- 15278 4898

Abstract— According to the International Technology Roadmap for Semiconductors (ITRS 2001), embedded memories will continue to dominate the increasing system on chips (SoC) content in the next years. Therefore the memory yield will have a dramatically impact on the overall defect-per-million (DPM) level, hence on the overall SoC yield. Meeting a high memory yield requires understanding memory designs, modeling their faulty behaviors in the presence of defects, designing adequate tests and diagnosis strategies as well as efficient repair schemes. This paper presents the state of art in memory testing including fault modeling, test design, Built-In-Self-Test (BIST) and Built-In-Self-Repair (BISR). Further research challenges and opportunities are discussed in enabling testing (embedded) memories.

Keywords— Memory fault models, test/diagnosis memory algorithms, built-in-self test, built-in-self repair.

I. INTRODUCTION

According to the 2001 ITRS, today's system on chips (SoCs) are moving from logic dominant chips to memory dominant devices in order to deal with today's and future application requirements. Figure 1 shows how the dominating logic (about 64% in 1999) is changing in dominating memory (more than 52% today). In addition, SoCs are expected to embed memories of increasing sizes, e.g. 256Mbits and more. As a result, the overall SoC yield will be dominated by the memory yield. Due to the fact that memory yield decreases with the increasing amount of memory, the overall yield may become unacceptable, unless special measures have been taken. The bottom curve in Figure 2 shows how the increase in the memory sizes can impact the yield. For instance, the yield of 20Mbits of embedded memory is about 35%; the example assumes a chips size of 12mm x 12mm, with a memory defect density of 0.4/square-inch and logic defect density of 0.4/square-inch, in 0.13 micron technology. To ensure/enhance an optimal yield level (upper curve in Figure 2), embedded memories must have the repair capabilities. Detecting the faulty dies only is no longer sufficient for SoCs; diagnosis and repair algorithms are often required. A repair algorithm uses a

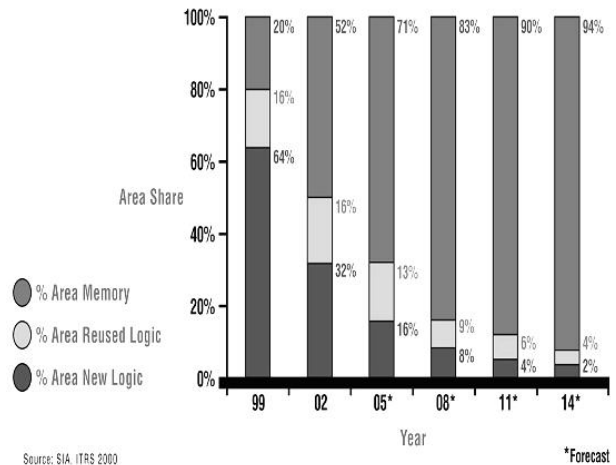


Fig. 1. The future of embedded memory

binary fault bit-map as its input. Such bit-map has to be produced based on the used test/diagnosis patterns to catch/locate defect cells. For embedded memories, test pattern(s) is generally programmed inside the BIST engine due to the lack of the controllability of their inputs and the observability of their outputs. The memory test patterns have to guarantee very high defect coverage, otherwise defect memories may escape; hence increasing the DPM level. The quality of the test patterns in terms of the defect coverage (and test length) strongly depends on the used fault models. New memory scaling technologies and processes are introducing new defects that were unknown in the past, and therefore new fault models are emerging.

This all clarifies that the challenges in embedded SoC memory testing will be driven by the following:

- **Fault modeling:** New fault models should be established in order to deal with the new defects introduced by today and future (deep-submicron) technologies.
- **Test algorithm design:** Optimal test/diagnosis algorithms to guarantee high defect coverage for the new memory technologies en reduce the DPM level.
- **BIST:** The only solution that allows at-speed testing for embedded memories.
- **BISR:** Combining BIST with efficient and low cost repair schemes in order to improve the yield and system reliability.

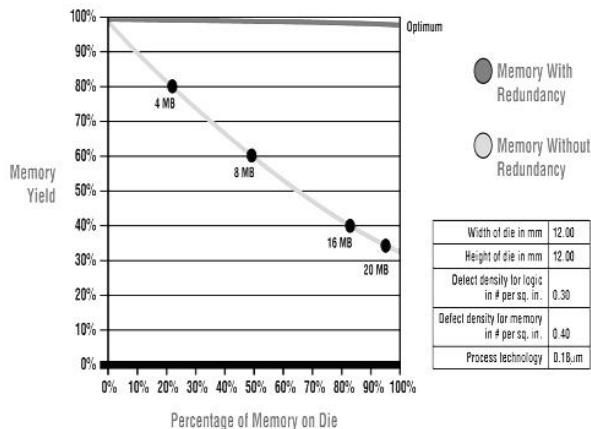


Fig. 2. Memory sizes versus yield

In the rest of the paper, the state of art of each of the above issues will be discussed, and the research challenges will be highlighted.

II. FAULT MODELING

During early 1980's many functional fault models for memories have been introduced. The advantage of such model is that the fault coverage of a certain test can be proven, while the test time is usually of order $O(n)$; i.e., linear with the size of the memory cell. Some well important models introduced in that time are [32]: Stuck-at-Fault and Address-Decoder-Faults. These models are abstract models and are not based on any real memory design and/or real defects. To reflect the faulty behavior of the real defects in real design, Inductive Fault Analysis (IFA) was introduced. IFA Allows for the establishment of the fault models based on simulated defects in real physical layout of the design. In addition, IFA is capable of determining the occurrence probability and the importance of each fault models. The result was that new fault models were introduced [12]: State-Coupling Fault and Data-Retention Fault. Late in 1990's, experimental results based on DPM screening of large number of tests applied to a large number of chips indicated that many detected faults cannot be explained with the well known models [27], [33], which suggested the existence of additional faults. This stimulated the introduction of new fault models, based on defect injection and SPICE simulation [1], [3], [13]: Read Destructive Fault, Write Disturb Fault, Transition Coupling Fault, Read Destructive Coupling Fault, etc.

The published work on memory fault modeling described above focuses on faults sensitized by performing *at most one operation*. Memory faults sensitized

by performing at the most one operation are referred as *static faults*.

A. Dynamic Fault Models

Recent published work reveals the existence of another class of faults in the new memory technologies. It was shown that another kind of faulty behavior can take place in the absence of static faults [4], [15], [16]. This faulty behavior, called *dynamic fault*, requires *more than one operation* to be performed *sequentially* in time in order to be sensitized. For example, a write 1 operation followed immediately by a read 1 operation will cause the cell to flip to 0; however, if only a single write 1 or a single read 1, or a read 1 which is not immediately applied after write 1 operation is performed, then the cell will not flip. [4] observed the existence of dynamic faults in the new embedded DRAMs based on defect injection and SPICE simulation. [15] observed the presence of dynamic faults in embedded caches of Pentium processor during a detailed analysis of the DPM screening results of a large number of test patterns. [16] showed the importance of dynamic faults for new SRAM technologies by analyzing DMP screening results of Intel and STMicroelectronics products, and concluded that current and future SRAMs need to consider dynamic faults testability or leave substantial DPM on the table.

The tests currently used in the industry have been designed to target static faults and therefore may not detect/diagnose dynamic faults. The dynamic fault class, which has been ignored in the past, is now becoming important and has to be taken into consideration. This sets a new direction for further research on memory fault modeling. Items like the following need to be worked out:

- Establishing the complete fault space, the fault framework and the fault models for dynamic faults.
- Validation based on defect injection and SPICE simulation.
- IFA in order to determine the occurrence probabilities and the importance of each introduced fault models, and provide better understanding of the underlying defects causing the dynamic faults.

B. Other Fault Modeling Aspects

Another special property of memories is that they have signals with a very high fan out. Examples of such signals are bit lines, word lines and address decoder pre-select lines. As the memories grow in size and operate on faster speeds, the lines carrying those signals will have, in addition to a high load, also a

high parasitic capacitance. This all makes them very sensitive for delay and timing related faults because of their capacitive coupling with other signals, power and ground lines. Moreover, the significance of the resistance opens is considered to increase in recent and future technologies due to the presence of many, long interconnections and the growing number of metal layers and vias. As the partial resistive opens behave as delay and time related faults, these faults will become more important in the deep-submicron technologies.

Another aspect that has to be taken into consideration for the deep submicron technologies is the soft errors. The increased operation speed and noise margin reduction that accompany the technological scaling, are reducing continuously the reliability of new technologies memories face to the various internal sources of noise. This process is now approaching a point where it will be infeasible to produce memories that are free from these effects. The nanometer ICs are becoming so sensitive that even sea level radiation will introduce unacceptable soft errors [26]. Designing soft error tolerant circuits is the only way to follow the technological scaling. Among the most efficient techniques are error detecting and error correcting codes. This will not only detect and correct the soft errors and the new failure, but also compensate for the possible incomplete test/diagnosis fault coverage.

Other considerations for fault modeling for new technologies are (but not limited to):

- Transistor Short channel effect: lowering the threshold voltage may make the drain leakage contribution significant.
- Cross talk effect and noise from power lines.
- The impact of process variation.

III. TEST ALGORITHM DESIGN

Memory tests and fault detection have experienced a long evolutionary process. The early tests (typically before the 1980's) can be classified as the *Ad-Hoc* tests because of the absence of formal fault models and proofs. Tests as Scan, Galpat and Walking 1/0 [6] belong to this class. They have further the property that for a given fault coverage, the test time was very long (excluded Scan), typically of order of $O(n^2)$, which made them very uneconomical for larger memories.

After the introduction of fault models during the early of 1980's, march tests became the dominant type of tests. The advantages of march tests lay in two facts. First, the fault coverage of the considered/known models could be mathematically proven, although one could not have any idea about the corre-

lation between the models and the defects in the real chips. Second, the test time for march tests were usually linear with the size of the memory, which make them acceptable from industrial point of view. Some well known march tests that have been shown to be efficient are: Mats+ [25], March C- [23], PMOVI [11], IFA 13n [12], etc. As new fault models have been introduced end 1990's, other new march tests have been developed to deal with them. Examples of such tests are March SR [13] and March SS [14].

Conventional memory test algorithms are basically designed to detect static functional faults (that are likely to occur) in order to determine if the chip is defective or not; in other words, they are pass/fail tests for static faults. As it have been shown in the previous section, the importance of dynamic faults increases with the new memory technologies. In addition, the shrinking technology will be a source of previously unknown defects/faults. The traditional tests are becoming thus insufficient/ inadequate for the today's and the future high speed memories. Therefore, new appropriate test algorithms have to be developed. On the other hand, as the memories become to represent the significant part of the SoC and dominant the overall yield, memory fault diagnosis becomes very important. Diagnosis techniques play a key role during the rapid development of semiconductor memories for catching design and/or manufacturing errors and failures; hence improving the yield. Although diagnosis has been widely used for memories, it is considered an expensive process due to long test times and complex fault/failure analysis procedure. Efficient diagnosis algorithms will benefit the industry and will play a more important role in the future as the SoC market grows.

Considering the current situation in test algorithm design and today's industry needs, can be concluded that new test/diagnosis algorithms still need to be developed; such algorithms have to take into consideration the following practical issues:

- Optimality in terms of time complexity in order to reduce the overall test time.
- Regularity and symmetric structure as possible so that the self-test circuit can use minimum of silicon area.
- High defect coverage and diagnosis capability in order to increase the repair capabilities and the overall yield.
- Appropriate stress combinations (voltage, temperature, timing, etc) that facilitate the detection of marginal faults.

IV. BUILT-IN-SELF TEST (BIST)

It is difficult to test an embedded memory simply by applying test patterns directly to the chip's I/O pins, because the embedded memory's address, data, and control signals are usually not directly accessible through the I/O pins. Large (and expensive) external tests cannot provide the needed test stimulus to enable high speed nor high quality tests [22]. BIST is therefore the only practical and cost-effective solution for embedded SoC memories.

BIST engine, no matter what kind, can use pseudo-random or deterministic patterns. A Pseudo-random pattern is basically very helpful to test logic circuits. A memory, however, has regular structure and requires application of regular test patterns. In the early days of BIST, it was not unusual to see pseudo-random techniques applied to memory [9], however these days this approach is hardly used due to its low fault coverage [2]. BISTs based on pseudo-random patterns are utilized on an occasional basis in the characterization of a design. Their design is mainly based on a linear feedback shift register (LFSR), which employs series of latches and XOR gates to implement a certain primitive polynomial [32].

BISTs based on deterministic patterns are dominant for testing memories today. Deterministic patterns means that patterns are generated according to specified pre-determined values (e.g., march tests). For the design of such BISTs, two techniques are mainly used: state machines and micro-codes.

A state machine BIST can generate a single simple pattern or complex suite of patterns [31]. This BIST is generally used in the industry to generate a single pattern (e.g., a single march test). However, a better memory test solution requires a suite of patterns; this makes the design of the state machine complex. A state machine BIST, as the name indicates, can exist in a number of states, which are group of latches from very few to several hundreds [7]. The major limitation of such BIST lays in its quite restricted flexibility. Modifying the patterns require changing the BIST design.

Micro-code BIST is a programmable BIST [19], [29], and therefore does not have limited flexibility. As new technologies are introducing faults that were previously unknown, new fault models can become evident during the fabrication. The BIST should be thus modifiable to include new patterns covering the new faults. The micro-code BIST is the most flexible of all self-test structures. The memory patterns can be easy

modified to assist in the characterization of the new memory designs. In addition, programmable BIST can be used in both manufacturing and in a system environment. Due to the flexibility property, different patterns can be utilized depending on the applications.

For BIST design, in addition to minimizing the performance penalty introduced for normal memory operations, an important additional criterion is to minimize the area and the pin overhead. Embedded memories are usually of different widths, and are much smaller than stand-alone memories, resulting in high BIST overhead. BIST technology combines several different areas; e.g., fault modeling and test design. As the memories are increasing in size and as SoCs are including several memories of different sizes, with different access protocols and timing, the BIST technology is facing several practical issues like:

- Minimizing BIST overhead in both silicon area and routing.
- Selecting the proper number of BIST controllers to be used (i.e., choosing the proper clustering for multiple arrays for BIST controller hardware sharing).
- Supporting diagnosis capabilities.
- Fulfilling the power budget constraints.
- Supporting different kind memories (single-port, multi-port).

V. BUILT-IN-SELF-REPAIR (BISR)

As the complexity and the size of the embedded memories keep increasing, several challenges appear in the scene of memory repair in order to improve the overall yield. Using redundancy or spare elements is one of the most known used ways to improve the yield.

The most traditional manner for performing memory repair is using external test and repair. It starts first with applying memory test algorithms (can be on chip (BIST) or by ATE) and collecting the response in order to build the bit-map to be stored in a large capture memory on the ATE via limited I/O bandwidth; of course a high quality diagnosis algorithms have a critical role in this step. The off-chip failure bit-map used to record faults requires a large memory. The ATE software then uses the failed bit-map to determine the best way to allocate redundant elements to replace the defective elements and to generate the reconfiguration data. An optimal repair algorithm has been shown to be NP hard [10] and therefore requires a long execution time. The generated reconfiguration data is thereafter used for hard repair (laser/ electri-

cal fusing). The fusing equipments will program the memory by blowing the fuses corresponding to the defective memory cells. The repaired memory has finally to be retested in order to ensure that the repair was successful.

The limitations of external repair lay in different factors. It relies on the extensive ATE, which makes the test cost of the chip very high (about 40% of the overall manufacturing cost of the semiconductor chip [34]). Since the external method relies on general-redundancy allocation algorithms, it also has limited repair efficiency. In addition, it only provides a limited I/O bandwidth for sending the large failed bit-map from the embedded memories to ATE. Furthermore, laser repair is often very expensive and some times continuous periodic field repair is desired [35].

To deal with the above limitations and reduce the overall manufacturing cost, the memory has to be made self-repairable. This will be achieved by expanding the embedded test resources even further to include a storage repair data and soft configuration mechanisms. In other words, BIST resources for future embedded memories will require to move beyond fault detection to include failed bit-map diagnosis, redundancy analysis and self repair. Once the failed bit-map is generated, based on diagnosis algorithm, the repair efficiency depends mainly on the type and the amount of redundancy, and on the allocation algorithms. The most common types of redundancy are: row redundancy, column redundancy, or block redundancy. Many allocation algorithms for BISR solutions have been proposed where the failure information does not need to be stored off-chip. Generally speaking, they are simple solutions.

In [8] a self-repairing structure for high hierarchical ultra-large memory chips is introduced. The repair scheme uses spare rows with the memory blocks at the lowest level of hierarchy and block redundancy at the top level. Although global redundancy can repair a great variety of faults, because of its greater inherent flexibility, it suffers from a higher area overhead.

A BISR for high density SRAMs using only spare columns and a greedy algorithm for allocation is proposed in [20]. The repair of defective circuits occurs autonomously without external stimulus (e.g., laser repair) by mapping redundancy columns via multiplexors to functionally replace the defective cells. The repair takes place immediately once a defective cell is found. Therefore no failure information is needed to be stored.

In [5] a solution is proposed based on combination of spare rows and columns. The scheme is based on divide-and-conquer strategy, where the memory is partitioned into several small identical segments; each segment is repaired independent of the others. However, the technique limits the numbers of spares to only one spare row pair and one spare column per each memory block in order to make the hardware realization practical and feasible.

A two-dimensional (i.e., spare rows and spare columns) repair allocation algorithm using a simple heuristics based on a finite state machine is presented in [24]. The scheme requires the storage (on chip) of only the final addresses to be repaired and the repair information, which is supplied to the laser repair equipment as direct programming data. The evaluation of the technique with up to five spare rows and five spare columns reveals that the scheme does not always guarantee 100% reparability of the memory, and therefore the allocation is not always possible.

The BISR analyzer suggested in [18] is based on an exhaustive search of all possible repair solutions for embedded DRAMs; the method therefore guarantee 100% detection ability of the repairable chips. The analyzer uses m spare rows and n spare columns per block (where $m \geq 2$ and $n \geq 2$) and search all possible solution $C(n+m,n) = (n+m)!/(m!n!)$. If multiple solutions found, the analyzer chooses the solution with the minimum number of used spares. Although the scheme improves the analysis speed drastically, requires no external failure bit, reduces test time (due to at speed testing), and provide a repair solution if it exist, it suffers from practical limitations. Its implementation is mainly based on CAM (Content Addressable Memory). The size of the required CAM increases exponentially with the number of spare rows and columns $(m+n)$; i.e., $(n+m)*C(m+n,n)$ cells.

All known repair algorithms are not optimal and practically have restricted applications, for smaller number of spare elements only. New repair schemes for arbitrary number of spares yet guaranteeing the solution have to be found while considering practical issues like:

- Dealing with complicated memory structures.
- Low hardware cost.
- Test time reduction. This can be achieved by using BIST and BISR at speed.
- On fly repair.
- Applicability to different types of memories.
- Repair methodologies.

To generate a high quality test strategy for new (embedded) memory technology, a through procedure must be pursued. First the memory design has to be well understood. The circuits need to be investigated not only in the way they are expected to operate, but also in the way each of the circuits operates in the presence of various defects. These defective and faulty operations need to be mapped into fault models. Thereafter, the best test patterns can be developed. Since no single test can achieve an acceptable DPM level, a suite of test patterns needs to be used. Understanding the design, fault models and tests is required in order to prevent shipping defective parts. Redundancy and repair goes beyond that and are required to guarantee adequate yield on the vast majority of memories. The design, fault modeling and test design have to be revisited in light of redundancy. Redundancy algorithms need to be generated to allocate each redundancy dimension to the appropriate fails, whereby maximizing the yield. Once the design, fault models, test patterns, and redundancy algorithms are understood, the correct built-in-self testing scheme can be designed while achieving a very low DPM level and a very high overall yield.

REFERENCES

- [1] D. Adams and E.S. Cooley, "Analysis of Deceptive Read Destructive Memory Fault Model and Recommended Testing", *In Proc. Of North Atlantic Test Workshop*, 1999
- [2] D. Adams, High Performance Memory Testing, Kluwer Academic Publisher, ISBN: 1-4020-7255-4, 2003.
- [3] Z. Al-Ars and A.J. van de Goor, "Impact of Memory Cell Array Bridges on the Faulty Behavior in Embedded DRAMs", *In Proc. of ATS*, pp. 282-289, 2000.
- [4] 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. Of DATE*, pp. 401-406, 2001
- [5] D.K. Bhavsar, "An Algorithm for Row-Column Self Repair of RAMs and Its Implementation in the Alpha 21264", *In Proc of ITC*, pp. 311-317, 1999.
- [6] M.A. Breuer and A.D. Friedman, "Diagnosis and Reliable Design of Digital Systems", Computer Science Press, Woodland Hills, CA, USA, 1976.
- [7] P. Camurati, et.al, "Industrial BIST for Embedded RAMs", *IEEE Design & Test of Computers*, Vol. 12, No. 3, pp. 86-95, 1995.
- [8] T. Chen and G. Sunada, "Design Self-Testing and Self-Repairing Structure for High Hierarchical Ultra-Large Capacity Memory Chips", *IEEE Trans. On VLSI*, Vol. 1, No. 2, pp. 88-97, June 1993.
- [9] R. David, A. Fuentes and B Courtois, "Random Patterns Testing Versus Deterministic Testing of RAMs", *In IEEE Trans. On Computers*, Vol. 38, No., 5, pp. 637-650, 1989.
- [10] J.R. Day, "A Fault-Driven Comprehensive Redundancy Algorithm", *IEEE Design & Test of Computers*, Vol 2, No. 2, pp. 35-44, June 1985.
- [11] J.H. De Jonge and A.J. Smeulders, "Moving Inversions Test Pattern is Thorough, Yet Speedy", *In Comp. Design*, pp. 169-173, 1979.
- [12] R. Dekker, et al., "A Realistic Fault Model and Test Algorithms for Static Random Access Memories", *IEEE Trans. on Computers CAD*, Vol. C9, No. 6, pp. 567-572, 1990.
- [13] S. Hamdioui, A.J. van de Goor, "Experimental Analysis of Spot Defects in SRAMs: Realistic Fault Models and Tests", *In Proc. of ATS*, pp. 131-138, 2000.
- [14] S. Hamdioui, A.J. van de Goor and M. Rodgers, "March SS: A Test for All Static Simple RAM Faults", *In Proc. of IEEE International Workshop on Memory Technology, Design, and Testing*, pp. 95-100, Bendor, France, 2002
- [15] S. Hamdioui, Z. Al-ars and A.J. van de Goor, "Testing Static and Dynamic Faults in Random Access Memories", *In Proc. of IEEE VLSI Test Symposium*, pp. 395-400, 2002.
- [16] S. Hamdioui, et. al, "Importance of Dynamic Faults for New SRAM Technologies", *In IEEE Proc. Of European Test Workshop*, pp. 29-34, 2003.
- [17] T. Kawagoe, et.al, "A Built-in Self-Repair Analyzer (CRESTA) for embedded DRAMs", *In Proc of ITC*, pp. 567-573, 2000
- [18] H. Koike, T. Takeshima and M. Takada, "A BIST Scheme Using Microprogram ROM for Large Capacity Memories", *In Proc. of IEEE Int. Test Conference*, pp. 815-822, 1990.
- [19] I. Kim, et. al, "Built In Self Repair For Embedded High Density SRAMs", *In Proc of ITC*, pp. 1112-1119, 1998
- [20] S.Y. Kuo and W.K. Fuchs, "Efficient Spare Allocation for Reconfigurable Arrays", *IEEE Design & Test of Computers*, Vol 4, No. 1, pp. 24-31, February 1987.
- [21] R. Mookerjee, "Segmentation: A Technique for Adapting High-Performance Logic ATE to Test High-Density, High-Speed SRAMs", *In Rec. of MTDT*, pp. 120-124, 1993
- [22] M. Marinescu, "Simple and Efficient Algorithms for Functional RAM Testing", *In Proc. of IEEE International Test Conference*, pp. 236-239, 1982.
- [23] S. Nakahara, et.al, "Built-In Self-Test for GHz Embedded SRAMs Using Flexible Patterns Generator and New Repair Algorithm", *In Proc. of ITC*, pp. 301-307, 1999
- [24] R. Nair, "An Optimal Algorithm for Testing Stuck-at Faults Random Access Memories", *IEEE Trans. on Computers*, Vol. C-28, No. 3, pp. 258-261, 1979.
- [25] M. Nicolaidis, "Design for Soft Error Robustness To Rescue Deep Submicron Scaling", *In Proc. Of ITC*, pp. 1140, 1998.
- [26] I. Schanstra and A.J. van de Goor, "Industrial evaluation of Stress Combinations for March Tests Applied to SRAMs", *In Proc. IEEE Int. Test Conference*, pp. 983-992, 1999.
- [27] P.G. Shepard, et.al, "Programmable Built-In Self-Test Method and Controller for Arrays", USA patent Number 5.633.877, 1997
- [28] J. van Sas, et.al, "BIST For Embedded Static RAMs with Coverage Calculatins", *In Proc of International Test Conference*, pp. 339-347, 1993.
- [29] A.J. van de Goor, "Testing Semiconductor Memories, Theory and Practice," ComTex Publishing, Gouda, Netherlands, 1998.
- [30] A.J. van de Goor and J. de Neef, "Industrial Evaluation of DRAMs Tests", *In Proc. of Design Automation and Test in Europe*, pp. 623-630, March 1999.
- [31] Y. Zorian, S. Dey and M. Rodgers, "Test of Future System-on Chip", *In Proc. Of ITC*, pp.392-398, 2000.