Testing Embedded Memories: A Survey

Said Hamdioui

Computer Engineering Lab
Delft Univeristy of Technology
Mekelweg 4, 2628CD, Delft
The Netherlands
S.Hamdioui@tudelft.nl
http://www.ce.ewi.tudelft.nl/hamdioui/

Abstract. According to the International Technology Roadmap for Semiconductors, embedded memories will continue to dominate the increasing system on chips (SoCs) content in the future, approaching 90% in in some cases. Therefore, the memory yield and quality will have a dramatic impact on the overall SoC cost and outgoing product quality. Meeting a high memory yield and quality requires understanding memory designs, modeling their faulty behaviors in appropriate and accurate way, designing adequate tests and diagnosis strategies as well as efficient Design-for-Testability and Built-In-Self-Test (BIST) schemes. This paper presents the state of art in memory testing including fault modeling, test design and BIST. Further research challenges and opportunities are discussed in enabling testing (embedded) memories in the nano-era.

Keywords: Memory Test, Fault Modeling, test algorithm design, MBIST.

1 Introduction

The semiconductor memory development over the years can be classified in three phases [1]: (a) the stand-alone, (b) memory integrated with logic and (c) scaled embedded memory. In the first phase, typically from about 1980 to 1990, the ideal MOS memory was a standardized stand-alone part, with its small cell size, good array efficiency, adequate performance, noise and soft error resistance, and met an external I/O standard. In the second phase, occurred from 1990 to 2000, where memories began to be integrated onto the logic chip; however, embedded DRAM and Flash were hindered by the historical divergence of the memory and logic technologies. In the third phase, from 2000 on, the era of true embedded memory has begun. 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 have been 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

A. Kučera et al. (Eds.): MEMICS 2012, LNCS 7721, pp. 32–42, 2013.

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 ('gigascale') and highly integrated technologies ('nano-scale').

The cost of memory testing increases with every generation of new memory chips [2]. Precise *fault modeling* to design *efficient tests*, in order to keep the test cost and test time within economically acceptable limits, is therefore essential. The quality of the tests, in terms of defect/fault coverage, is strongly dependent on the used fault models. Therefore, fault models reflecting the real defects of the new memory technologies are crucial for developing higher quality test algorithms and therefore providing products with low Defect-Per-Million (DPM) level driven by the market.

It is difficult, if not impossible, to test an embedded memory simply by applying test patters 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. Therefore, *Memory Built-In-Self Test (MBIST)* is used for memory testing. The basic philosophy behind the MBIST technique is: "let the hardware test itself "; i.e. enhance the functionality of the memory to facilitate self-test. Large (and expensive) external tests cannot provide the needed test stimuli to enable high speed, nor high quality tests [3]. BIST therefore is the only practical and cost-effective solution for embedded SoC memories.

This paper addresses the state-of the art of the three major aspects related to embedded memory testing; these are fault modeling, test algorithm design and MBIST. In addition, future challenges and trends will be covered.

2 Memory Fault Modeling

As already mentioned, the quality of fault models used to develop test algorithms is very crucial in providing high outgoing product quality measured in DPM level. The concept of memory fault model appeared first in early 1980's. Many fault models have been developed targeting different fault behaviors. Memory fault models can be classified into two classes: (a) *static faults*, developed mainly between 1980 and 2000, and (d) *dynamic faults*, have been developed since begin 2000.

Static Faults

During the early 1980's many memory functional fault models have been introduced, allowing the fault coverage of a certain test to be provable. Some important fault models introduced in that time are [4,5,6]: Stuck-at-Faults and Address-Decoder-Faults. These are abstract fault 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 designs, Inductive Fault Analysis (IFA) was introduced. IFA allows for the establishment of the fault models based on simulated defects at the physical layout level of the design. In addition, IFA is capable of determining the occurrence probability and the importance of each fault model. The result was that new fault models were introduced [7]: State-Coupling Fault and Data-Retention Fault. In the late 1990's, experimental results based on DPM screening of a large number of tests applied to a large number of memory chips indicated that many detected faults cannot be explained with the well-known fault models [8,9], which suggested the existence of additional faults. This stimulated the introduction of new fault models, based on defect injection and SPICE simulation [10,11]: Read Destructive Fault, Write Disturb Fault, Transition Coupling Fault, Read Destructive Coupling Fault, etc.

All the memory fault modeling described above focuses on faults sensitized by performing *at most one operation.* For instance, Read Destructive Coupling Fault is sensitized by applying a read operation to the victim cell, while the aggressor cell is put in a certain state (i.e., the required number of operations is 1). Memory faults sensitized by performing at most one operation are referred to as *static faults.*

Dynamic Faults

Many work published since early 2000 have revealed the existence and the importance 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 [12]-[21]. This faulty behavior has been attributed to *dynamic faults*, which require *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. [12] observed the existence of dynamic faults in the new embedded DRAMs based on defect injection and SPICE simulation. [13,14] observed the presence of dynamic faults in embedded caches of Pentium processors during a detailed analysis of the DPM screening results of a large number of tests. [15,16] showed the importance of dynamic faults for new SRAM technologies by analyzing DPM screening results of Intel and STMicroelectronics products, and concluded that current and future SRAMs tests need to consider dynamic faults or leave substantial DPM on the table. [17] showed the existence of dynamic memory cell array faults for SRAMs using defect injection and circuit simulation. The work of [18,19] proved the existing of dynamic faults in the peripheral circuits of a memory (such as sense amplifiers, pre-charge circuits, etc.), while the work of [20,21] showed the existence of this class of faults in the address decoders. Due to the importance of covering dynamic faults in order to realize the required product quality (as it has been show by measured data [14]-[16], [22]), tests for such faults become an integral part of memory test programs used within industry [23,24,25]. Dynamic faults are becoming even more important for the future technologies [26].

3 Test Algorithm Design

Tests and fault detection for semiconductor memories have experienced a long evolutionary process, starting before 1980's. Overall, memory test algorithms can be classified into three classes: (a) Ad-hoc tests, (b) March tests, and (b) Fault primitive based tests; they are explained next.

Ad-hoc Tests: The are the early tests (typically before the 1980's). They are classified as Ad-Hoc because of the absence of formal fault models and proofs. Tests as Scan, Checkerboard, Galpat and Walking 1/0 [5,27] belong to this class. They have further the property of either having a low fault coverage (as it is the case for Scan and Checkerboard) or requiring a very long test time (as it is the case for Galpat and Walking), which make them very uneconomical for larger memories.

March Tests: 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 correlation between the models and the defects in the real chips. Second, the test time for march tests is typically linear with the size of the memory, which made them acceptable from an industrial point of view. Some well-known march tests, that have been shown to be efficient, are: Mats+ [28], March C- [29], PMOVI [30], IFA 13n [7], etc.

Fault Primitive Based Tests: As new fault models have been introduced in the late 1990's, based on defect injection and SPICE simulation, other new tests have been developed to deal with them. In addition, the concept of Fault Primitive (FP) was introduced to better describe a *single* fault behavior [31]; it made it easier to analyze the failure mechanisms in deep-sub micron technology and to develop realistic fault models and thereafter optimal test algorithms using defect-oriented testing. This approach allows for the realization of high defect coverage as the algorithm can be optimized for each design/layout. Some of the introduced FP based tests are targeting static faults; examples are March SS [32] and March MSS[33] for detecting all possible static faults, March SR [11] for detecting faults realistic for the design under consideration, etc. However, most of the developed FP based tests are targeting dynamic faults, including memory cell array faults [13]-[16], peripheral circuits faults [18,19] and address decoder faults [20,21,34].

It is worth noting that the state-of-the art in memory fault modeling and test design typically assumes the presence of a single fault at a time; memory tests assuming the presence of multiple faults at a time have got limited attention in the past [35,36] and they seems to become more important with further technology scaling [37]. Fault modeling and test design have to consider not only the presence of a single defect/fault at a time, but also 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 sensitizes it; e.g., due to a partial open defect that creates a small delay. However, a fault can be fully sensitized (i.e., becomes strong) when two (or more) weak faults are sensitized *simultaneously* since their fault effects can be additive.

4 Memory Built-In-Self Test

As already mentioned, it is difficult to test an embedded memory simply by applying test patters 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. Therefore, BIST is the only practical and cost-effective solution for embedded SoC memories.

BIST engines, no matter what kind, can use pseudo-random or deterministic test patterns/algorithm. A pseudo-random pattern is basically very helpful to test logic circuits. A memory, however, has a regular structure and typically requires the application of regular and deterministic test patterns as those discussed in the previous section. In the early days of BIST (typically before 1990's), it was not unusual to see pseudo-random techniques applied to memory [38,39]. However, this approach has been hardly used from 1990 on due to its low fault coverage.

MBIST based on deterministic patterns is dominant for testing memories today. Deterministic patterns means that the patters are generated according to specified predetermined values (such as march tests). When implementing MBIST engines, trade-offs are made depending on: (a) the number of supported algorithms, (b) the flexibility of the MBIST engine (in order to cope with the unexpected), (c) the implementation speed, and (d) the area overhead. In addition, when MBIST performs tests, memory accesses have to be done at-speed using Back-to-Back memory cycles in order to detect dynamic faults [14]-[25]. Systems require large, high speed memories, while the technology scaling exhibits a large spread in implementation parameters, resulting in speed related (dynamic) faults.

Many papers have been published on deterministic based MBIST; examples are [40]- [50]. Let's assume that every memory test algorithm can be described using an extended notation of March algorithms [27], including the non-linear algorithms such as Galpat and Walking 1/0. A march test consists of a finite sequence of March Elements (MEs) [4]; a march element is a finite sequence of operations applied to every cell in the memory before proceeding to the next cell. Based on the level at which memory algorithms are specified, MBIST engines can be classified into four classes.

Algorithm Based MBST: Only a single (or few) algorithms can be specified; they are implemented in hardware using a state machine [40,41]. This MBIST is generally used in 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. The major limitation of algorithm based MBIST lays in its quite restricted flexibility; modifying the patterns requires changing the MBIST design. This MBIST implementation was used in the early days of MBIST, where the fault behavior was still considered simple and mainly static, while the implementation cost was very important. Performing at speed-testing was not a crucial issue for such MBIST class.

March Element Based MBIST: Applying an algorithm consists of successively scanning-in each of the algorithm's MEs, together with its algorithm stress combination. In order to be able to perform memory tests at-speed, it is sufficient that each of the individual MEs is performed at-speed [42,45,49,50]. For this MBIST class, not the algorithms are hard-wired, but only the MEs. Recent publications [49,50] show that such MBIST class can provide higher flexibility at extremely small command memory, which makes this MBIST class very attractive for embedded applications whereby the tests have to be stored within the MBIST engine. Moreover, all information required to support an at-speed implementation is contained in the specified MEs. Hence, the required detailed information to control the memory, such as whether a read or write has to be performed, can be decoded from the ME prior to its application. This prevent the complex implementation (costly hardware) which typically uses complex schemes such as pipelining and prefetching to apply at speed-testing.

ME operations Based MBIST: In this case, at speed of only the operations within a single ME can be performed; elapsed time between MEs is not critical. Typically each ME can use a certain stress combination (such as data-background). Therefore this class can be easy implemented using two registers (one for ME and one for the stress combination) and a register controlled state-machine; the two register can be scanned via a low speed tester. Because of its low cost implementation, such MBIST is very popular within the industry. Note that ME operations based MBIST requires more test time than March Element Based MBIST; in the latter all MEs are hard-wired and no external scanning is needed for MEs to perform the tests.

Individual Operations Based MBIST: This class of MBIST engines allows for the specification of algorithms by specifying each of the algorithm operations. Obviously, this can be very flexible. However, the implementation cost is effectively determined by the size of the algorithm memory and the supported addressing scheme(s) [46]-[48], [51]. The more algorithms and addressing schemes, the more hardware overhead. In addition, it typically explores expensive prefetching and pipelining techniques to perform at speed-testing, which makes the implementation costly. This class is mainly suitable for higher end products.

5 Future Challenges

This section addresses some major challenges and trends wrt embedded memory testing and gives some research directions. First, the technology technology threats due to continue scaling will be discussed. Thereafter, the business pressure will be covered. Finally, the requirements for future memory test solutions will be described.

5.1 Technology Threats

Progressive technology scaling, as tracked by the ITRS and encapsulated by Moore's law, 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, allowing the integration of a wider variety of functions. However, it is well recognized that many challenges are emerging:

- *Extreme variations*: The increasing variability in device characteristics and its impact on the overall quality and reliability represent major challenges to scaling and integration for future nanotechnology generations [26] (cross talk, interferences, leakages, Vth mismatch, degraded Read/Write margins, ...). 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, parametric and weak faults rather than hard and permanent faults; hence causing more reliability problems than quality problems [52]-[54] .
- *Reduced voltages*: Although the supply voltage is not scaling at the same scale as the technology, the reduced voltages are contributing to the emergence of many new failure mechanisms that may impact either the quality and/or the reliability of memories; examples are: reduced signal strength, reduced SNM (higher soft error rates) and increased sensitivity to delay (parametric) faults.
- *Speed related faults*: The increasing clk speed of each memory generation poses new challenges. For instance, a 10ps delay for a memory running at 100MHz is only 1% of the Clk speed. However, this is 10% for a memory running at 1GHz! What is used to be known as marginal delays can cause timing failures for today and future technologies. Hence testing for at-speed related faults is becoming a must. Not to mention that other emerging failure mechanisms also contribute to speed related faults.
- *Wearout*: The aggressive technology scaling does not only cause new failure mechanisms, but also increases the level of transient errors (during device lifetime) and reduces the device lifetime, causing major reliability challenges.

5.2 Business Pressure

The continuous increases in SoC complexity in general and the simultaneous higher competitive semiconductor industry pose many business pressure challenges.

- *Higher customer requirements*: irrespective of the business pressure, customers always require a higher product quality, lower cost and higher reliable chips. Higher customer satisfaction requires therefore *higher fault coverage*, even for unknown faults. Hence, large set of tests and stresses are needed, making test cost higher (reducing benefits).

- *Shorter time-to-volume (TTV)/market*: TTV consist of two parts: (a) design time and (b) production ramp-up time [1]. IP reuse is a common practice used to reduce the design time. Reducing production ramp-up time is what is causing the real bottleneck. Traditionally, one had sufficient learning time for test cost and DPM level reduction, starting at low volume with a large number of tests used to detect, analyze and correct yield problems. Today, and due to time-to-market pressure, the time-to-market is reduced causing a severe reduction in the learning curve. Due to the very short to no learning curve, understanding of all faults for each new technology is impossible. Hence, the test program may provide a *low fault coverage*, which contradicts the customer requirements.

5.3 Requirements for Future Test Solutions

Given the challenges mentioned above, the future solutions have to provide answers to both the customer requirements (higher quality, reliability and fault coverage for all emerging faults) and the short time-to-market (higher yield). Today's solutions are going in the following direction:

- *Manufacturing Test*: Use effective test set. Development of new tests may require new approaches. To optimize cost v quality (wrt time-to-volume), this test may not detect all faults (economically undetectable faults escape). In addition, use a programmable BIST at module level to enhance shorter time to market
- *In field Test*: Use Error Correction codes (ECC) to compensate for incomplete fault coverage and to detect soft errors and new (unexpected) failures. In addition, use dynamic Built-In-Self-Repair to maintain the effectives of ECC and increase yield and product lifetime.

6 Conclusion

This paper has discussed three major aspects related to embedded memory testing and has provided some future challenges. The approach based on single defect a time causing a *strong fault*, on which the traditional fault modeling and test design are based, may need refinement as memories in nano-era may suffer from different small disturbances (*weak faults*) at the same time; these weak faults can together create a strong fault if sensitize *simultaneously* during the application. Therefore a new memory test paradigm may be needed.

References

1. Marinissen, E.J., et al.: Challenges in Embedded Memory Design and Test. In: Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (2005)
2. Inoue, M., et al.: A New Test Evaluation Chip for Lower Cost Memory Tests. IEEE Design and Test of Computers 10(1), 15–19 (1993)

3. Mookerjee, R.: Segmentation: A Technique for Adapting High-Performance Logic ATE to Test High-Density, High-Speed SRAMs. In: IEEE Workshop on Memory Test, pp. 120–124 (1993)
4. Suk, D.S., Reddy, S.M.: A March Test for Functional Faults in Semiconductors Random-Access Memories. IEEE Transactions on Computers C-30(12), 982–985 (1981)
5. Abadir, M.S., Reghbati, J.K.: Functional Testing of Semiconductor Random Access Memories. ACM Computer Surveys 15(3), 175–198 (1983)
6. Papachistou, C.A., Saghal, N.B.: An Improved Method for Detecting Functional Faults in Random-Access Memories. IEEE Trans. on Computers C-34(2), 110–116 (1985)
7. Dekker, R., et al.: A Realistic Fault Model and Test Algorithms for Static Random Access Memories. IEEE Trans. on Computers 9(6), 567–572 (1990)
8. Schanstra, I., van de Goor, A.J.: Industrial evaluation of Stress Combinations for March Tests Applied to SRAMs. In: Proc. IEEE Int. Test Conference, pp. 983–992 (1999)
9. van de Goor, A.J., de Neef, J.: Industrial Evaluation of DRAMs Tests. In: Proc. of Design Automation and Test in Europe, pp. 623–630 (1999)
10. Adams, D., Cooley, E.S.: Analysis of Deceptive Read Destructive Memory Fault Model and Recommended Testing. In: Proc. of IEEE NATW (1999)
11. Hamdioui, S., van de Goor, A.J.: Experimental Analysis of Spot Defects in SRAMs: Realistic Fault Models and Tests. In: Proc. of Ninth Asian Test Symposium, pp. 131–138 (2000)
12. Al-Ars, Z., van de Goor, A.J.: Static and Dynamic Behavior of Memory Cell Array Opens and Shorts in Embedded DRAMs. In: Proc. of Design Automation and Test in Europe, pp. 401–406 (2001)
13. Hamdioui, S., Al-ars, Z., van de Goor, A.J.: Testing Static and Dynamic Faults in Random Access Memories. In: Proc. of IEEE VLSI Test Symposium, pp. 395–400 (2002)
14. Hamdioui, S., van de Goor, A.J., Reyes, J.R., Rodgers, M.: Memory Test Experiment: Industrial Results and Data. IEE Proceedings of Computers and Digital Techniques 153(1), 1–8 (2006)
15. Hamdioui, S., et al.: Importance of Dynamic Faults for New SRAM Technologies. In: Proc. of European Test Workshop, pp. 29–34 (2003)
16. Hamdioui, S., Wadsworth, R., Reyes, J.D., Van De Goor, A.J.: Memory Fault Modeling Trends: A Case Study. Journal of Electronic Testing 20(3), 245–255 (2004)
17. Dilillo, L., et al.: Dynamic read destructive fault in embedded-SRAMs: analysis and march test solution. In: Proc. Ninth IEEE European Test Symposium, pp. 140–145 (2004)
18. Van de Goor, A.J., Hamdioui, S., Wadsworth, R.: Detecting faults in the peripheral circuits and an evaluation of SRAM tests. In: Proc. of Inter. Test Conference, pp. 114–123 (2004)
19. Dilillo, L., et al.: Resistive-Open Defect Influence in SRAM Pre-Charge Circuit: Characterization and Analysis. In: 10th European Test Symposium on IEEE ETS 2005 (2005)
20. Hamdioui, S., Al-Ars, Z., van de Goor, A.J.: Opens and Delay Faults in CMOS RAM Address Decoders. IEEE Trans. on Computers 55(11), 1630–1639 (2006)
21. Dilillo, L., et al.: ADOFs and Resistive-ADOFs in SRAM Address Decoders: Test Conditions and March Solutions. Jour of Electronic Testing: Theory and Applications 22(3), 287–296 (2006)

22. Hamdioui, S., Al-Ars, Z., Jimenez, J., Calero, J.: PPM Reduction on Embedded Memories in System on Chip. In: IEEE Proc. of European Test Symposium, Freiburg, Germany, pp. 85–90 (May 2007)
23. Al-Ars, Z., Hamdioui, S., Gaydadjiev, G.N., Vassiliadis, S.: Test Set Development for Cache Memory in Modern Microprocessors. IEEE Trans. Very Large Scale Integration (VLSI) Systems 16(6), 725–732 (2008)
24. Powell, T., Kumar, A., Rayhawk, J., Mukherjee, N.: Chasing Subtle Embedded RAM Defects for Nanometer Technologies. In: Proc. of the IEEE Int. Test Conf., paper 33.4, vol. 23(5) (October 2007)
25. Mukherjee, N., Pogiel, A., Rajski, J., Tyszer, J.: High Volume Diagnosis in Memory BIST Based on Compressed Failure Data. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems 29, 441–453 (2010)
26. Bhavnagarwala, A., et al.: The semiconductor industry in 2025. In: IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), pp. 534–535 (2010)
27. van de Goor, A.J.: Testing Semiconductor Memories, Theory and Practice, 2nd edn. ComTex Publishing, Gouda (1998)
28. Nair, R.: An Optimal Algorithm for Testing Stuck-at Faults Random Access Memories. IEEE Trans. on Computers C-28(3), 258–261 (1979)
29. Marinescu, M.: Simple and Efficient Algorithms for Functional RAM Testing. In: Proc. of IEEE International Test Conference, pp. 236–239 (1982)
30. De Jonge, J.H., Smeulders, A.J.: Moving Inversions Test Pattern is Thorough, Yet Speedy. In: Comp. Design, pp. 169–173 (1979)
31. van de Goor, A.J., Al-Ars, Z.: Functional fault models: A formal notation and taxonomy. In: Proc. IEEE VLSI Test Symp., pp. 281–289 (2000)
32. Hamdioui, S., van de Goor, A.J., Rodgers, M.: March SS: A Test for All Static Simple RAM Faults. In: Proc. of IEEE International Workshop on Memory Technology, Design, and Testing, Bendor, France, pp. 95–100 (2002)
33. Harutunvan, G., Vardanian, V.A., Zorian, Y.: Minimal March tests for unlinked static faults in random. In: Proc. of IEEE VLSI Test Symposium, pp. 53–59 (2005)
34. van de Goor, A.J., Hamdioui, S., Gaydadjiev, G.N., Al-Ars, Z.: New Algorithms for Address Decoder Delay Faults and Bit Line Imbalance Faults. In: 18th IEEE Asian Test Symposium, Taichung, Taiwan, pp. 391–397 (November 2009)
35. van de Goor, A.J., et al.: March LA: A test for linked memory faults. In: Eur. Design Test Conf., p. 627 (1999)
36. Hamdioui, S., et al.: Linked faults in random access memories: concept, fault models, test algorithms, and industrial results. IEEE Trans. on CAD of Integrated Circuits and Systems 23(5), 737–757 (2004)
37. Hamdioui, S., et al.: A New Test Paradigm for Semiconductor Memories in the Nano-Era. In: Proc. of Asian Test Symposium, pp. 347–352 (2011)
38. McAnney, et al.: Random Testing for Stuck-At Storage Cells in an Embedded Memory. In: Proc. of Intern. Test Conference, pp. 157–166 (1984)
39. David, R., Fuentes, A., Courtois, B.: Random Patterns Testing Versus Deterministic Testing of RAMs. IEEE Trans. on Computers 5, 637–650 (1989)
40. Dekker, R., Beenker, F., Thijssen, L.: Realistic built-in self-test for static RAMs. Design & Test of Computers 6(1), 26–34 (1989)
41. Dreibelbis, J.H., Hedberg, E.L., Petrovic, J.G.: Built-In Self-Test for Integrated Circuits, US Patent, Number 5,173,906 (December 22, 1992)
42. Zarrineh, K., et al.: A new framework for generationg optimal March tests for memeory arrays. In: Proc. of the Int. Test Conf., pp. 73–82 (1998, 2001)

43. Powell, T.J., et al.: BIST for Deep Submicron ASIC Memories with High Performance Application. In: Proc. of the IEEE Int. Test Conf., pp. 386–392 (2003)
44. Appello, D., et al.: Exploiting Programmable BIST For The Diagnosis of Embedded Memory Cores. In: Int. Test Conference, p. 379 (2003)
45. Aitken, R.C.: A Modular Wrapper Enabling High Speed BIST and Repair for Small Wide Memories. In: Proc. of the IEEE Int. Test Conf., paper 35.2, pp. 997–1005 (2004)
46. Du, X., Mukherjee, N., Cheng, T.M.: Full-Speed Field-Programmable Memory BIST Architecture. In: Proc. of the IEEE Int. Test Conf., paper 45.3 (2005)
47. Du, X., Mukherjee, N., Cheng, W.-T., Reddy, S.M.: A Field-Programmable Memory BIST Architecture Supporting Algorithms and Multiple Nested Loops. In: Proc. of the Asian Test Symposium, paper 45.3 (2006)
48. van de Goor, A.J., Jung, C., Gaydadjiev, G.: Low-cost, Flexible SRAM MBIST Engine. In: IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems (April 2010)
49. van de Goor, A.J., Hamdioui, S., Gaydadjiev, G., Alars, Z.: Generic March Element Based Memory Built-In Self-Test, Dutch Patent Application; Filing Number NL 2004407, Filed date (January 2010)
50. van de Goor, A.J., Hamdioui, S., Kukner, H.: Generic, orthogonal and low-cost March Element based memory BIST. In: Inter. Test Conference, pp. 1–10 (2011)
51. Khare, J.B., Shah, A.B., Raman, A., Rayas, G.: Embedded Memory Field returns - Trials and Tribulations. In: Proc. IEE Int. Test Conf., Paper 26.3 (2006)
52. Borkar, S.: Design and Test Challenges for 32 nm and Beyond. Keynote speech at IEEE International Test Conference, p. 13 (2009)
53. Hamdioui, S., Al-Ars, Z., Mhamdi, L., Gaydadjiev, G.N., Vassiliadis, S.: Trends in Tests and Failure Mechanisms in Deep Sub-micron Technologies. In: IEEE Proc. of Int. Conference on Design and Test of Integrated Systems in Nanoscale Technology, pp. 216–221 (September 2006)
54. Vermeulen, B., Hora, C., Kruseman, B., Marinissen, E.J., van Rijsinge, R.: Trends in Testing Integrated Circuits. In: Proc. IEEE Int'l Test Conf., pp. 688–697 (2004)