Implementation Of Fault Awareness For Memory Lbist Architecture

G.Mounika 1 , D.Rambabu2

1. PG Scholar, Department of ECE, Ramachandra College of Engineering and Technology, JNTUK, A.P...Mail id–mounikag411@gmail.com

2.Associate Professor, Department of ECE, Ramachandra College of Engineering and Technology, , JNTUK, A.P...Mail id-drambabuece@gmail.com

***ABSTRACT*:** *Testing cost is one of the major contributors to the manufacturing cost of integrated circuits. Logic Built-In Self-Test (LBIST) offers test cost reduction in terms of using smaller and cheaper ATE, test data volume reduction due to on-chip test pattern generation, test time reduction due to at-speed test pattern application. However, it is difficult to reach a sufficient test coverage with affordable area overhead using LBIST. Also, excessive power dissipation during test due to the random nature of LBIST patterns causes yield-decreasing problems such as IR-drop and overheating. In order to increase test coverage of LBIST, we propose to use on chip circuitry deterministic test patterns. The problem of excessive test power dissipation is addressed with a scan partitioning algorithm which reduces capture power for delay fault LBIST. Our experimental results show that, on average, the presented method reduces average capture power and peak capture power with less than drop in the transition fault coverage. By comparing the proposed algorithm to the original scan partitioning, we show that the proposed method is able to achieve higher capture power reduction with less fault coverage drop.*

KEY WORDS: Logic Built-In Self-Test (LBIST), built-in self-test (BIST), automated test equipment (ATE). Fault Periodicity Table (FPT), Test Algorithm Template (TAT)

I.INTRODUCTION

Historically, till very recent times the performance improvements of integrated circuits (ICs) have depended mainly on scaling planar device structures. Although nowadays scaling of transistors and other circuit elements is becoming more difficult approaching fundamental physical limitations on device size, it is still one of the most important dependency factors for the performance improvements. Two new factors have been appeared very recently which, at least, can be considered for the performance improvements. One of them is connected with the change of the IC basic element – transistor and creation of a multi-gate device or multiple gate field-effect transistor which incorporates more than one gate into a single device. Currently the most investigated in the literature such device is known under the term FinFET. It was coined by University of California, Berkeley researchers to describe a non-planar (3D) transistor. Due to growing leakage and short-channel problems of conventional planar transistors, it is not possible to continue with Moore's Law by further scaling down the feature sizes of these planar transistors. The second factor is 3D ICs which can address interconnect latency which is also beginning to limit IC performance: Vertical connections allow shorter connections.

Expansion of numbers of interconnects through compact vertical connections. 3D ICs promise to overcome barriers in inter-connect scaling by leveraging fast, dense inter-die vias, thereby providing an opportunity for continued higher performance using CMOS. In addition, 3D ICs also enable the integration of heterogeneous fabrication processes on the same chip to make the form factor more compact. As we discussed the transistors are evolving from planar to FinFET and very recently a new candidate for next generation transistors is introduced called Gate-All-Around (GAA). In GAA transistor the gate covers the transistor channel on all sides. As for memories, those are also evolving starting from 2D memories and in nowadays technologies 3D memories are started to be widely used. In addition to evolution of transistors and memories, the complexity of System on Chips (SoCs) is also actively evolving. If the initial SoCs contained several memories with one centralized built-in self-test scheme then current SoCs consist of multiple sub-chips where each sub-chip may contain thousands of memories and multiple built-in self-test schemes.

Modern memory built-in self-test (BIST) controllers come either with hardwired standard test algorithm or with a programmability option. In the first case, a user cannot modify or add his/her own test algorithm to the hardwired BIST. Thus, such a test algorithm is typically not optimized for a novel or proprietary memory design; whereas, the programmable option uses an architecture that usually requires an externally accessible register (Test Algorithm Register) of predefined format for storing a micro-program that will perform a given test algorithm.

Typically, the programmable BIST controller provides enough flexibility for test algorithm definition. At the same time, the test algorithm is composed of special test mechanisms (test operations, background patterns and addressing methods) which are usually hardwired in the BIST infrastructure. In this context, the programmability of BIST means that only the existing test mechanisms can be used when programming a test algorithm. This limitation specifically impacts the flexibility of test coverage and test time. In some cases, this limitation can even lead to the impossibility of programming the required test algorithm. Afterwards, solutions with programmability of separate test mechanisms are proposed. However, all the solutions mentioned above do not have a common backbone which will allow considering these different issues within a unified infrastructure. In this paper, it is proposed to describe all the observed fault regularity and periodicity rules in form of a special Fault Periodicity Table (FPT). Each column of FPT corresponds to a fault nature (e.g., number of cells that the corresponding fault is involved with), which can be associated with a variety of different test mechanisms covering the fault. Each row of FPT corresponds to a fault family determined by the complexity of fault sensitization. Fault symmetry is also taken into account in the FPT. The FPT not only allows building of a generic BIST architecture that supports programmability of test algorithms without limitations mentioned above. It also allows detection of new faults which arise in the field after manufacturing within the same BIST. If these faults can be predicted beforehand then due to regularity of the FPT it is possible to include them into the range of faults covered by the BIST architecture and to detect them further via programmability.

II.RELATED WORK

In this paper, a new solution for building the mentioned BIST infrastructure is proposed which is based on multidimensional prediction mechanism. It is based on a notion of periodicity and regularity of faults and test algorithms, and their interdependence. These notions are represented in a form of regularity and periodicity rules described in Section III. This is considered as a basis for building a generic BIST architecture. Some explanations are shown below to clarify the idea. We consider the known classification of faults, based on the following factors:

 Complexity of fault sensitization, i.e., number of operations required to activate the fault;

 Number of cells that a fault is involved with.

Following this classification, the faults can be grouped into different classes. The number of classes and faults inside them increase along with technology shrinking. We have done a systematic investigation of the evolution of these fault classes and their detection algorithms, covering a broad range of manufacturing technologies from 90nm to 7nm. We have discovered different types of new faults, such as special classes of dynamic faults, process variation and random telegraph noise induced faults. Results of this investigation led to determination of some regularity and periodicity rules, which exist for the evolution. Moreover, the known property of symmetry, which means that each fault usually has its twin, was developed further and a special symmetry measure for March test algorithms was introduced.

The dependency between symmetry measure and BIST optimization is discussed, as well as a new representation of March test algorithms is introduced called canonical view of March tests. Canonical view represents all similar fragments of a March test algorithm in a standard form comprised of a basic sequence of March elements and a sequence of its transformations for each March element. This means that if two consecutive parts in a March test have the same structure but are different only by data backgrounds, addressing directions, data polarities or addressing modes, then we can describe only the first part and for the second part describe only its difference compared with the first part. This allows to use less bits to store a test algorithm in Test Algorithm Register (TAR) of a BIST compared to storing the original view of a March test. The experiments showed that the BIST area overhead can be reduced by 30-40% with negligible time overhead when using canonical view of March tests.

Based on the property of fault and test algorithm symmetry, a new efficient method was proposed to generate symmetric March test algorithms. The method is based on the observation that almost all known minimum or efficient March test algorithms are symmetric. Term “minimum” in the text refers to having minimum length among all the possible solutions. Based on this, an efficient test algorithm March LSD of complexity 75N is generated for detection of all static faults, two-operation dynamic faults and all possible links between them. Another direction of FPT usage is the following. There can be faults that are not realistic in the current technology, but can be predicted as realistic in the future technologies. These faults can also be reflected in the BIST architecture. This becomes urgent in cases when it is planned to port a given design, including the BIST infrastructure, to a new technology basis without making design changes. Not all dependencies and regularities of the FPT are already known. Particularly, during our investigation, several periodical dependencies and regularities that were unknown before were found, justified and

included in the FPT. They include also a symmetry reflection in test mechanisms.

III. EXISTED SYSTEM

Test data volume has been becoming larger as the size and the complexity of digital designs continue to grow. Therefore manufacturing test comprises a growing fraction of a modern VLSI device's total cost. LBIST is the technique of designing additional hardware and software features into LSI and allow them to perform self-testing, i.e., testing of their own operation using their own circuits, thereby reducing dependence on external automated test equipment (ATE).

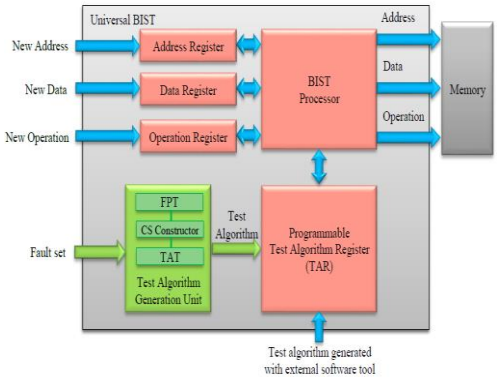


Fig. 1. EXISTED SYSTEM

The above figure (1) shows the architecture of existed system. The devices used in this system are register, BIST Processor, test algorithm register, memory. Two types of methods can be used to reduce capture power in scan-based test. The first type reduces capture power by making test patterns power friendly. These methods can only be applied to external test as they utilize the large portion of unspecified bits in ATPG patterns. The second type is scan partitioning (also called segmented scan), in which only a subset of scan cells are activated in shift and capture cycles. Ideally, the power consumption is reduced by a factor of n if the scan consists of n partitions. As a price to pay for the reduced capture power, the test application time is multiplied by the number of partitions.

It is also possible that test coverage decreases, because the former captured scan cells might affect the values captured by the latter ones. This is a phenomenon known as capture violation. The technique presented in copes with capture violation problem by allowing partitions to be captured in different order. The scan partitioning problem can be modelled as minimization of the number of violation edges in the S-graph. An S-graph (or data dependency graph of a sequential circuit is a directed graph, of which the vertex set is the set of ip-ops in the circuit, and an edge from vertex u to v represents a combinational path from ip-op u to ip-op. An efficient S-graph partitioning algorithm based on mixed integer programming is presented. For LBIST, reducing power consumption normally involves modification of the test structures. The gating logic is used to partially mask scan cell activities. A low power BIST scheme based on circuit partitioning is presented. A substitute for an LFSR is introduced in to generate power-aware pseudo-random patterns. The scan partitioning method based on controllability/observability analysis is presented. However, all of the above methods that reduce capture power for LBIST target stuck at fault model. Their applicability and efficiency for delay fault model is unknown.

IV. PROPOSED SYSTEM

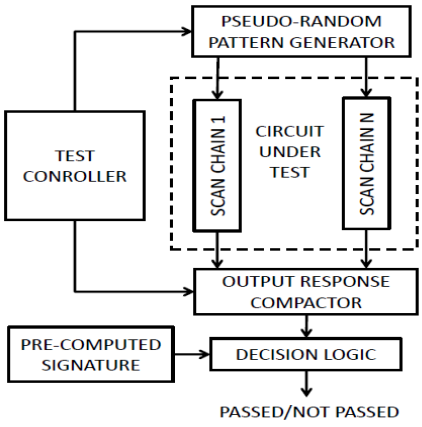


Fig. 2. PROPOSED SYSTEM

The above figure (2) shows the architecure of proposed system. The devices used in this system are test controller, pre-

computed signayyure, psedo random pattern generator, descision logic and comparator. Automatic Test Pattern Generation (ATPG) is a technique used to end a set of test vectors to allow an Automatic Test Equipment (ATE) to check whether a circuit is malfunctioning. A full scan design can be considered as purely combinational. For such designs, ATPG algorithms such as the D-algorithm, path-oriented decision making, and fan out-oriented test generation are available. The test set for a partial scan design can be generated using sequential ATPG algorithm. Fault models are used to model manufacturing defects in integrated circuits. Two of the commonly used fault models are:
1. single stuck-at fault model, which assumes a single line in the circuit is stuck at a logic zero or a logic one;
2. Transition fault model, which assumes certain (rise or fall) transitions passing through a single line are delayed past the next clock tick.

For a certain fault model, there are a fixed number of faults in a circuit. Some of the faults may not be detectable. Given a set of test vectors, test coverage is defined as the number of detected faults divided by the number of detectable faults, while fault coverage is defined as the number of detected faults divided by the number of all faults. A set of test vectors generated by an ATPG algorithm is called a de- termini tic test set. Targeting certain test coverage, a deterministic test set with minimum number of test vectors is the minimal test set. 2.8.3 Logic Built-In Self-Test.

Built-In-Self-Test (BIST) attempts to reduce the raising complexity of external testing by incorporating test generation and response capture logic on-chip. On-chip circuitry usually works at a much higher frequency than a tester. So, by embedding the test pattern generator on chip, we can reduce test application time. In addition, by embedding the output response analyser on chip, we can reduce time to compute the circuit response. There are different types of BIST. Logic BIST (LBIST), on which we focus in this dissertation, is used for testing random digital logic. LBIST typically employs a Pseudo-Random Pattern Generator (PRPG) to generate test patterns that are applied to the circuit's internal scan chains, and an output response compactor for obtaining the compacted response of the circuit to these test patterns, called signature. Faults are detected by comparing the computed signature to the expected \good" signature.

There are several strategies for generating deterministic test patterns on chip. The first strategy is to store the deterministic patterns in an on-chip memory. The stored deterministic pattern can also be encoded first, which can later be decoded on-chip. Run-length codes record only the lengths of consecutive 0's (or 1's) to reduce the size of the stored patterns. Dictionary-based coding schemes partition the patterns into symbols, and use an additional dictionary to store all unique symbols so they can be accessed with their indices. The symbols can also be assigned variable-length code words, where the lengths are based on the frequency of the symbols' occurrences. LFSR reseeding is another strategy to generate deterministic test patterns based on the linearity of LFSRs. Since deterministic patterns contains a large number of unspecified bits, LFSR-generated sequences can be used to match" the deterministic sequences. When mismatch occurs, the LFSR is set to a pre-determined state, called a seed, and continues generating a matching sequence. The seeds are calculated by solving a system of linear equations. The probability that a pattern can be successfully encoded in a seed is over 99.9999% if the LFSR size is at least Smax + 20, where Smax is the maximum number of specified bits in a pattern. The seeds can be either stored in a memory, or

generated with a reseeding circuit on-chip. Several approaches are presented to increase to encoding efficiency, including using variable-length seeds and multiple polynomials and partial dynamic reseeding. The third approach transforms pseudo-random LFSR patterns into deterministic patterns through a mapping circuit that is put in between the LFSR and the circuit under test.

V. RESULTS

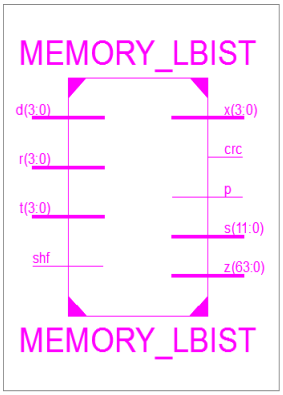


Fig. 3: RTL SCHEMATIC

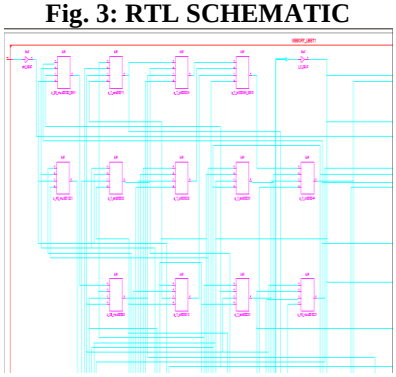


Fig. 4: TECHNOLOGY SCHEMATIC

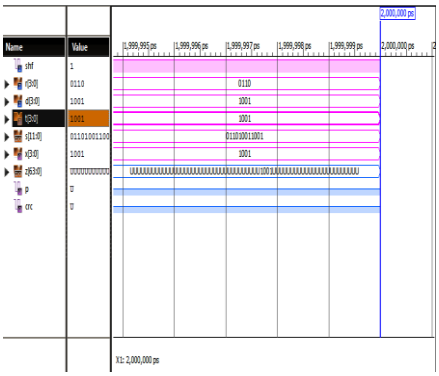
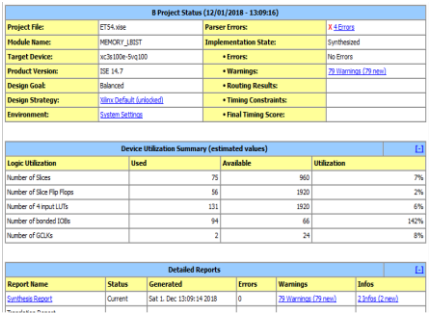


Fig. 5: OUTPUT WAVEFORM



Project File:	ET54.xise	Parser Errors:	X 4 Errors
Module Name:	MEMORY_LBIST	Implementation State:	Synthesized
Target Device:	xc3s100e-5vq100	• Errors:	No Errors
Product Version:	ISE 14.7	• Warnings:	79 Warnings (79 new)
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slices	75	960	7%
Number of Slice Flip Flops	56	1920	2%
Number of 4 input LUTs	131	1920	6%
Number of bonded IOBs	94	66	142%
Number of GCLKs	2	24	8%

Detailed Reports

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Sat 1. Dec 13:09:14 2018	0	79 Warnings (79 new)	2 Infos (2 new)

Fig. 6: REPORT

VI. CONCLUSION

Based on a systematic investigation of memory fault types and test algorithms, novel properties of periodicity and regularity are discovered and several rules reflecting these properties are proposed. Accordingly, special structures - Fault Periodicity Table (FPT) and Test Algorithm Template (TAT) are created and validated to immerse these rules. The FPT represents all memory faults in a single table, and the TAT produces efficient test algorithms. This infrastructure is intended to allow optimization of test time versus fault coverage, finding new faults, and predicting upcoming realistic faults. We continue our investigation on this domain and currently we see that the best choice depends on the dataset and the technique that is used. Nonetheless, there is still a simple take-away message. If the dataset is sparse, creation of a new model could work better. Otherwise reframing is a much better option. The use of plots in any case is recommended, in order to spot the contexts where a pair of approach and technique is better than the rest. Finally, resources are important criteria for how to proceed, as creating a model again and again may become infeasible for some applications, and reframing a single, versatile model may be a much better option in cost-effective terms.

VIII. REFERENCES

[1] P. Ranade, Y.-K. Choi, D. Ha, H. Takeuchi, T.-J. King, "Metal Gate Technology for Fully Depleted SOI CMOS", International AVS Conference on Microelectronics and Interfaces, 2003, pp. 131-133.

[2] H.-W. Cheng, Y. Li, "16-nm Multigate and Multifin MOSFET Device and SRAM Circuits", International Symposium on Next-Generation Electronics, 2010, pp. 32-35.

[3] M. Jurczak, N. Collaert, A. Veloso, T. Hoffmann, S. Biesemans, "Review of FinFET Technology", IEEE International SOI Conference, 2009, pp. 1-4.

[4] G.E. Moore, "Cramming More Components Onto Integrated Circuits", Electronics, Vol. 38, No. 8, 1965, pp. 114-117.

[5] R. S. Patti, "Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs", Proceedings of the IEEE, 2006, Vol. 94, No. 6, pp. 1214–1224.

[6] D.-I. Moon, S.-J. Choi, J.P. Duarte, Y.-K. Choi, "Investigation of Silicon Nanowire Gate-All-Around Junctionless Transistors Built on a Bulk Substrate", IEEE Transactions on Electron Devices, 2013, Vol. 60, No. 4, pp. 1355-1360.

[7] K. Park, J. Lee, S. Kang, "An Area Efficient Programmable Built-In Self-Test for Embedded Memories Using an Extended Address Counter", IEEE International SoC Design Conference, 2010, pp. 59-62.

[8] D. Youn, T. Kim, S. Park, "A microcode-based memory BIST implementing modified march algorithm", IEEE Asian Test Symposium, 2001, pp. 391-395.

[9] P. Bernardi, M. Grosso, M. Sonza Reorda, Y. Zhang, "A Programmable BIST for DRAM testing and diagnosis", IEEE International Test Conference, 2010