International Journal of Science Engineering and Advance Technology

On Fault Free Response Analysis For BIST

[1]Yogesh Singh and [2]Dr. Shafiqul Abidin
[1]Research Scholar, Dr. K. N. Modi University, Rajasthan, INDIA
[2] Professor, Department of Information Technology, HMR Institute of Technology & Management (Affiliated with GGSIP University), Hamidpur, Delhi, INDIA
singhyogesh2002@yahoo.co.in

Abstract- High levels of coverage of classical and non-classical faults require deterministic test sequences. In this paper, it is suggested that the deterministic test sequences be ordered in such a way that the fault-free output response is a trivial one that can be generated by simple on-chip circuitry, thereby obviating the need for test response compression.

***Index Terms*— TPG, aliasing, RA, Pseudo-Random, Deterministic.**

I. INTRODUCTION

The Built-In Self-Test property is defined as the provision of on-chip test pattern generation (TPG) and/or on-chip response analysis (RA). On the TPG side, besides the use of deterministic test patterns, pseudo-random testing has gained wide acceptance. For test response analysis, and besides the obvious bit-by-bit comparison, signature analysis is the technique most commonly used. The following table summarizes the disadvantages of different combinations of TPG and RA approaches.

TPG RA	Pseudo Random	Deterministic
Signature Analysis	Test sequence may not detect all faults. Even if all faults are detected, signature analysis introduces aliasing.	The deterministic nature of test sequence is lost because aliasing in signature analysis.
Bit-by-Bit Comparison	Test sequence may not detect all faults. On-chip implementation is not feasible.	On-chip implementation not feasible.

II. OVERVIEW PSEUDO RANDOM TEST

Pseudo-random test sequences have excessive lengths and they do not cover all stuck-at faults (the so-called 'random-resistant faults'). This prompted some researchers to 'modulate' the pseudo-random sequences in order to increase their fault coverage [1,2].Others have suggested that the circuit under test should be re-designed so that all random resistant faults are eliminated 3. The deficiencies of pseudo-random test sequences arc further aggravated when non-classical faults, such as CMOS stuck-open, are taken into consideration.

For test response evaluation, signature analysis uses an n-bit register to compress an *m*-bit output sequence, where *m* is usually much larger than *n*. In this case there are 2^m possible output sequences, only one of which is the correct one. Therefore, there are approximately 2^{m-n} output sequences that get compressed into the same signature, including the correct signature, i.e., there arc 2^{m-n} - 1 faulty output responses that get compressed into the correct signature (if n = 16 and m = 1024, then 2^{1008} - 1 faulty responses would escape detection if they occurred). This is clearly unacceptable.

III. FAULT DETECTION

To achieve high levels of fault detection, deterministic test pattern generation has to be used, especially for non-classical faults. The deterministic nature of the TPG will be wasted if signature analysis is used to compress the output response. On the other hand, simple bit-by-bit comparison cannot be considered for on-chip implementation because of the large amount of data required.

It is suggested in this paper, that test patterns be generated deterministically, but to get signature analysis, and bit-by-bit comparison, the test patterns should be ordered in such a way that the output response is a trivial one (i.e., easily generated by on-chip circuitry). The most trivial signal is a toggling one. Therefore, it is suggested that the test patterns be ordered in such a way that the output of the circuit under test is toggled from one clock cycle to the next one. A similar idea has been suggested in a recent publication [4], where the quotient output from a signature analyzer is made periodic to ease its monitoring in order to reduce aliasing errors.

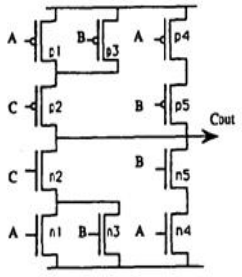
In the next section, we consider the testing of CMOS combinational circuits to show the motivation of the suggested approach, and following this, we address some of its practical issues.

IV. TESTING A CMOS CIRCUIT

A transistor stuck-open fault in a CMOS circuit usually requires a pair of vectors for its detection. For a pFET (nFET) stuck-open fault, the first, or initialization, vector must set the output of the gate containing the pFET (nFET) to 0 (1), and then the second, or test, vector must charge (discharge} the output through a path containing the pFET (nFET) under test. If the fault is present, the gate output will remain unchanged, whereas the output of a fault-free circuit will toggle.

If stuck-open faults f_a and f_b are detected by the two-pattern tests (T_{ao}, T_{a1}) and(T_{bo}, T_{b1}), respectively, and if $T_{a1} = T_{bo}$ then the sequence (T_{ao},T_{a1},T_{b1}) detects both faults. This optimization, which is possible when the initialization vector of some stuck-open fault is a test vector for another fault, will be much easier if some care is taken at the test pattern generation stage; since in most cases a stuck-open fault can have a number of initialization vectors.

Making use of the above observation repeatedly would yield a complete test sequence that produces a toggling fault-free output. Moreover, the presence of any stuck-open fault (or any other detectable fault, for that matter) would prevent, at least, one transition at the output. As an example, consider the testing of the circuit shown in Fig. 1.



Faults	Initialisation vector(s)	Test vector(s)
p1	011, 110, (101), (111)	010
p2	110, 101, 011, (111)	010, 100, (000)
p3	110, 101, (011), (111)	100
p4, p5	011, 101, (110), (111)	001
n1	001, 100, (010), (000)	101
n2	001, 010, 100, (000),	011, 101, (111)
n3	001, 010, (100), (000)	011
n4, n5	010, 100, (001), (000)	110

Figure 1. Example circuit and tests for all its stuck-open faults.

The possible initialization and test vectors for all stuck-open faults are also shown in Fig.1. The input vectors that are between brackets should be ignored if the tests are to remain valid under arbitrary circuit delays. The complete test sequence for this circuit is ABC= {011, 010, 110, 100, 101, 001, 011}. Note that the repetition of vector 011 is necessary since the initial state of the output node is assumed to be unknown.

V. PRACTICAL CONSIDERATION

The first criticism that one might find to the suggested approach is that an ordered deterministic test sequence requires a large area for the test pattern generator, as compared to an LFSR or a counter, for example. However, this may be offset by the simplicity of test response analysis. In a BIST environment, the analysis of the circuit response to a test sequence that toggles the output cannot be made any easier. Simple on-chip circuitry can be used on all nodes that need monitoring to perform bit-by-bit comparison with the fault-free response which, in this case, can be generated by simple on-chip hardware. This fault-free test response is the same for all circuits on the chip. It can be considered as simply an auxiliary clock signal, of half the frequency of the chip's clock, which is broadcast to all nodes that need monitoring. Furthermore, the problems of aliaising and information losses in methods that rely on data compression (signature analysis, one's count, etc.) do not exist for the approach suggested in this paper, which makes it particularly suitable in situations where high levels of fault coverage are mandatory.

Another problem arises when considering multiple output circuits: It is impossible for two distinct outputs of the same circuit to be toggled by the same input test sequence (unless they are complements of each other). A solution would be to sequentially test the outputs of a same circuit and, in order to reduce test time, test as many circuits as possible in parallel. In addition, the testing of two, or more, outputs may overlap in time for the portions of the test sequence where these outputs and toggled simultaneously. In the most complex case, when the outputs have some logic in common, the principles of circuit segmentation, as used for pseudo-exhaustive testing [5] can be of great help in reducing the test time.

REFERENCES

[1] E. Lindbloom, E. B. Eichelberger. and 0. P. Fiorenza "A Method for Generating Weighted Ranclom Test Patterns" IBM J. Res. and Develop., Vol.33, No. 2. March 1989, pp 1-19-161

[2] F. Brglez, G. Gloster and G.Kedem “Hardware-Based Weighted Random Pattern

[3] V. S. Iyengar and D. Brand "Synthesis of Pseudo-Random Testable Designs" Proc. IEEE International Test Conference, 1989,pp 501-508.

[4] S. Gupta, D. Pradhan and S. M. Reddy "Zero Aliaising Compression” Fault Tolerant Computing Symposium, 1990, pp 245-263

[5] J. G. Udell, Jr. and E. J. McCluskey "Pseudo-Exhaustive Test and Segmcutaliun: Formal Definitions and Extended Fault Coverage Results" Fault Tolerance Computing Symposium, 1989, pp 292-29