Directed-Binary Search in Logic BIST Diagnostics

Rohit Kapur
Synopsys Inc.

T. W. Williams
Synopsys Inc.

M. R. Mercer
Texas A&M

Abstract

Logic BIST is about to become a more main stream test method for IC testing. In some flows when a failure is encountered the IC is diagnosed to determine the cause of the failure. Diagnosing fails in Logic BIST is significantly different from that in a stored pattern test methodology. The first step is to determine the failing pattern or interval among the many patterns that were applied. Today this involves a binary search of the tests that were applied with Logic BIST. In this paper we improve on this binary search strategy to reduce the time taken to isolate the failing patterns by orders of magnitude.

Introduction

The most attractive feature of Logic BIST is that the test is packaged with the IC [1]. Such a feature that provides less visibility into the tests comes with its own set of problems when diagnostics is to be performed on the IC. The typical approach to diagnosing a failure in a network with Logic BIST is to first locate the first failing pattern or the first failing interval of patterns. This is performed by searching the Logic BIST test pattern space with a binary search technique [2] that iteratively divides the test pattern space in equal lengths in search of the failing pattern (or interval).

Directed-Binary Search

The traditional binary search procedure is best suited for situations where the failing patterns are evenly distributed across the entire test pattern space. The reality of the situation is that the fault coverage is not linearly increasing with pattern count but follows an exponential curve where the patterns in the beginning detect most of the faults and a long tail of the fault coverage curve is obtained over the bulk of the patterns. Binary search is not optimally suited for this scenario.

Figure 1 shows the fault coverage based search (Directed-Binary Search), which is directed to the left in terms of points in the pattern space. The first iteration in the search goes until the pattern that achieves 50% fault coverage. If the failure is not detected until this point the search would move to the right where the next iteration would step to the pattern number at which 75% fault coverage is achieved. In this figure it is assumed that the failure is observed at this point in the search and the next step would be to the left, splitting the failing interval to balance the fault coverage on either side of the search space.

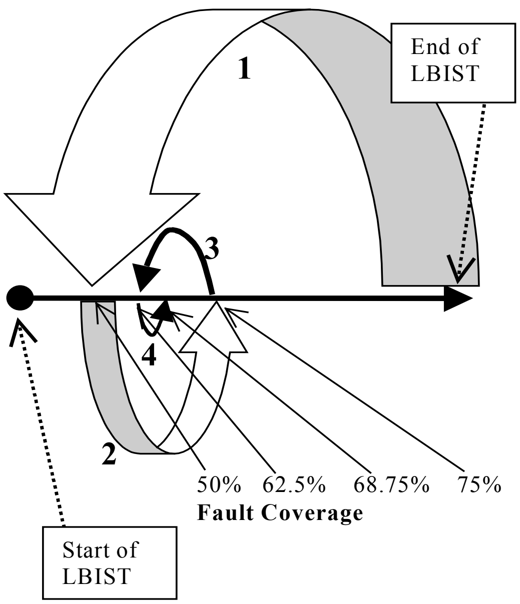

Figure 1: A binary search over fault coverage instead of patterns.

The expected length of this search procedure is of interest. Assuming that the failing test pattern is to be isolated to an interval of 256 test patterns the expected number of steps to locate the failing interval in a directed binary search process is 2.

References

[1] P. Bardell, W. H. McAnney and J. Savir, “Built-In Test for VLSI,” Wiley-Interscience, New York, 1987.

[2] M. Bell, G. Danaily, M. Howells and S. Pateras, “Bridging the Gap Between Embedded Test and ATE,” Proceedings of the International Test Conference, 2000, pp. 55-63.