TWO-INPUT OR/NOR CASCADE IN SPACE COMPRESSION IN VLSI

Sunil R. Das[1, 2], Satyendra N. Biswas[3], Altaf Hossain[2], and Mansour H. Assaf[2]
[1]Department of Computer Science, College of Arts and Sciences
Troy University, Montgomery, AL 36103 USA
[2]School of Information Technology and Engineering, Faculty of Engineering
University of Ottawa, Ottawa, ON K1N 6N5, Canada
[3]School of Engineering and Computer Science
Independent University, Dhaka 1329, Bangladesh

Introduction

A tremendous amount of complexity has been brought about to the test generation process of integrated circuits (ICs) due to very large scale integration (VLSI). With the unprecedented growth in electronics industry, the integration densities besides system complexities vastly increased and hence, the need for superior and more efficient methods of testing to ensure reliable operation of IC chips, the mainstay of today's many sophisticated devices and products, was intensely felt [1]–[7]. The very concept of testing, in general, has a broad applicability and finding highly effective testing techniques that ensure correct system performance is of immense practical significance. The conventional testing procedures of digital systems require application of test input patterns generated by a test pattern generator (TPG) to the circuit under test (CUT) and comparing the responses with known correct responses. For VLSI systems, because of higher storage requirements for the fault-free responses, the standard test processes become highly expensive and hence, alternate approaches are sought at minimizing the amount of required storage [6], [7]. Built-in self-testing (BIST) is a design philosophy that has the capability of solving many of the problems otherwise encountered in testing digital systems. It combines the concepts of built-in test (BIT) and self-test (ST) in *one*, termed BIST. In BIST, test generation, test application and response verification are all accomplished through built-in hardware. This allows different parts of an IC chip to be tested in parallel, thereby reducing the required testing time besides eliminating the need for external test equipment. A typical BIST architecture uses a TPG that sends its outputs to a CUT and output streams from the CUT are fed into a test data analyzer. A fault is detected if the test sequence is different from the response of the fault-free circuit. The test data analyzer is comprised of a response compaction unit (RCU), storage for the fault-free response of the CUT and a comparator. The extra logic representing the compression network, however, must be as simple as possible, to be easily embedded within the CUT and should not introduce signal delays to affect either the test execution time or normal functionality of the unit being tested. Additionally, signatures derived from faulty output responses and their corresponding fault-free signatures should not be the same, which unfortunately is not always the case. A fundamental problem with compression techniques is error masking or aliasing [7], which occurs when the signatures from faulty output responses map into the fault-free signatures. Aliasing causes loss of information, which in turn affects the test quality of BIST and thus, reduces the fault coverage (FC). Several approaches have been developed in the literature for computing the aliasing probability of which the exact computation is known to be NP-hard [7].

To reduce the amount of data represented by the fault-free and faulty CUT responses, data compression is used to create signatures (short binary sequences) from the CUT and its corresponding fault-free circuit. BIST techniques use pseudoexhaustive or exhaustive test patterns or on-chip storing of reduced or compact test sets. The RCU can be divided into two parts: a space compaction unit followed by a time compressor. In general, $\mathbf{P}$ input sequences coming from a CUT are fed into a space compressor, providing $\mathbf{L}$ output streams of bits such that $\mathbf{L} << \mathbf{P}$; most often, test responses are compressed into only *one* sequence ($\mathbf{L} = 1$). Space compression brings a solution to the problem of achieving high quality BIST of complex digital circuit chips without the necessity of monitoring a large number of internal test points, reducing testing time and area overhead by merger of test sequences coming from these internal test points into a single stream of bits. This single bit stream of length $\mathbf{H}$ is finally fed into a time compactor, yielding a shorter sequence of length $\mathbf{B}$ ($\mathbf{B} < \mathbf{H}$) at the output.

With this perspective in focus, the subject paper proposes technique for implementing aliasing-free BIST space support hardware with applications targeted specifically towards embedded cores-based system-on-chips (SOCs) [1]–[3], [7], extending some well-known concept of conventional switching theory, *viz.* that of compatibility relation as employed in the minimization of incomplete sequential machines, based on pairwise sequence mergeability, for detectable single stuck-line faults of the CUT. Taking advantage of mathematically sound selection criteria of aliasing-free merger of a pair of response outputs of the CUT by two-input OR/NOR logic, the paper achieves maximal compaction in the design, as was evident from extensive simulation runs conducted on the International Symposium on Circuits and Systems (ISCAS) 85 combinational and ISCAS 89 full-scan sequential benchmark circuits using fault simulation programs ATALANTA and FSIM. In view of space constraints, though, only partial results on simulation on ISCAS 85 combinational benchmark circuits using ATALANTA and FSIM are provided.

Development of Algorithms

The aliasing-free space compaction approach as developed in the paper is basically composed of two algorithms. The first stage of the proposed compression network is designed by implementing **ALGORITHM** $\mathcal{A}$, while the second and subsequent stages of the compaction network are developed by using **ALGORITHM** $\mathcal{B}$, based on the concept of pairwise fault detection (FD) compatibility and *conditional* FD compatibility [7].

Simulation Experience and Conclusions

To demonstrate the feasibility of the proposed zero-aliasing space compression technique, extensive simulations were conducted on different ISCAS 85 combinational and ISCAS 89 full-scan sequential benchmark circuits. In the design experimentation, we used ATALANTA (fault simulation program developed at the Virginia Polytechnic Institute and State University) as our test generation tool to generate the fault-free output sequences required to construct the space compactor circuits and to test the benchmark circuits using reduced test sets, accompanied with a random test session with FSIM fault simulation program that produces the necessary pseudorandom test vectors that detect most detectable single stuck-line faults for all the benchmark circuits. For each of the circuits, we computed FC before the compaction network, FC with the compression circuit, number of applied test vectors, simulation CPU time, estimates of hardware overhead, and maximal compaction ratio. For designing the first stage of the compactor, **ALGORITHM 𝒜** is run first on the original CUT to generate an FD compatible pair and then *conditional* FD compatible pairs of output lines for merger with two-input OR/NOR logic. After generating the first stage of the compactor, a new CUT is formed, which is used for generating the second and subsequent stages of the compression network, by following **ALGORITHM ℬ**. The process of merger has to be continued until either a single output or a minimal number of outputs are eventually obtained. The data presented in Fig. 1 list each of the ISCAS 85 combinational benchmark circuits under test, number of outputs of the CUT, number of outputs of the CUT and COMPRESSOR, compaction ratio and FC that was calculated as the baseline using ATALANTA with space compactors. The data presented in Fig. 2 are for the same circuits using FSIM, also after the compressor was added. Because of space constraints, only limited simulation results were shown here. As the test simulation data demonstrate, the results are pretty consistent with regard to the complexity of the CUT and theory. With successful implementation of this methodology, it might be possible someday to automate the overall design process used for the compactors.

References

1. Rajsuman, R. "*System-on-a-Chip: Design and Test*", Artech House, Boston, MA (2000).
2. Mourad, S. and Zorian, Y. "*Principles of Testing Electronic Systems*", John Wiley & Sons, NY (2000).
3. Williams, T. W. and Parker, K. P. Testing logic networks and design for testability. *Computer*, 21(1979) 9-21.
4. Thatte, S. M. and Abraham, J. A. Test generation for microprocessors. *IEEE Trans. Comput*., C-29(1980) 429-441.
5. Kuban, J. R. and Bruce, W. C. Self-testing the Motorola MC6804P2. *IEEE Des. Test Comput*., 1(1984) 33-41.
6. Das, S. R., Assaf, M. H., Petriu, E. M., and Sahinoglu, M. Aliasing-free compaction in testing cores-based system-on-chip (SOC) using compatibility of response data outputs. *Trans. SDPS*, 8(2004) 1-17.
7. Das, S. R. and Applegate, A. R. Test response compaction in VLSI BIST with array of two-input linear logic. *Proc. IASTED Int. Conf. on Modeling, Identification and Control*, (2011) 144-151.

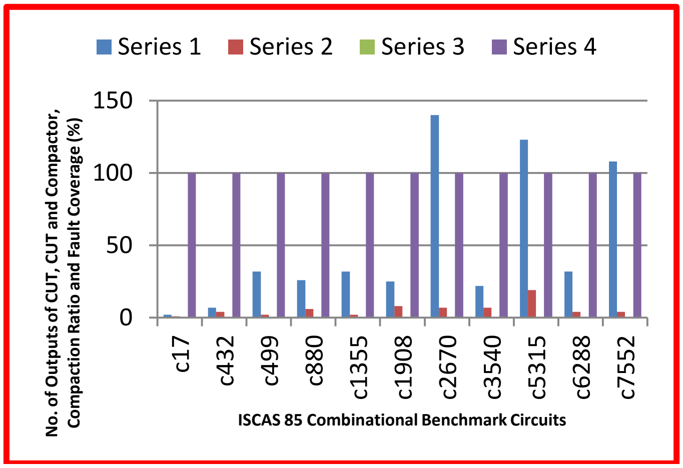

Fig. 1. Fault coverage of ISCAS 85 combinational benchmark circuits using ATALANTA with space compactors.

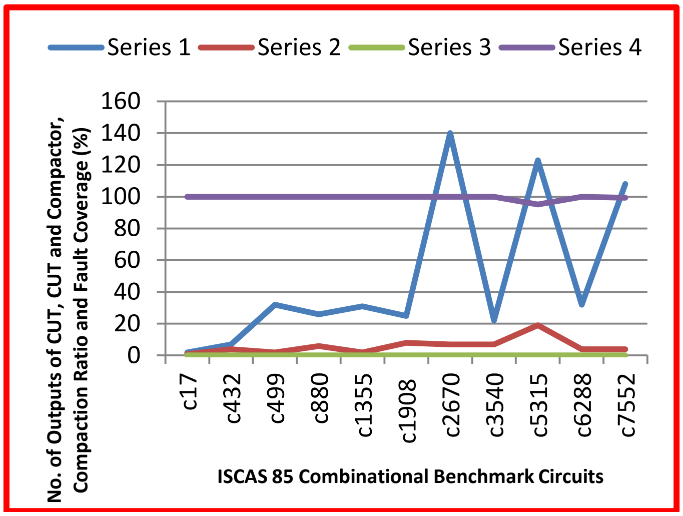

Fig. 2. Fault coverage of ISCAS 85 combinational benchmark circuits using FSIM with space compactors.

This research was supported in part by the Department of Computer Science, College of Arts and Sciences, Troy University, Montgomery, AL 36103.