Column-Matching BIST Exploiting Test Don't-Cares

Petr Fiser, Jan Hlavicka†, Hana Kubatova
Department of Computer Science and Engineering
Czech Technical University
Karlovo nam. 13, 121 35 Prague 2
e-mail: fiserp@fel.cvut.cz, kubatova@fel.cvut.cz

Abstract

We propose a new test-per-clock BIST method for combinational or full-scan circuits. Our aim is to design a combinational block transforming the LFSR code words into deterministic test patterns pre-computed by some ATPG tool. The proposed algorithm is an enhancement of a column matching method, in which the maximum of the output variables of the decoder is tried to be implemented as mere wires, thus without any logic. The enhancement consists in extending the use of the method for a test set containing don't cares. These don't cares allow us to reach a higher number of column matches, which significantly reduces the BIST logic.

1 Introduction

With a growing complexity of VLSI circuits their testing using external testers becomes extremely difficult. The internal logic of the circuits is then hardly accessible and the test length rapidly grows. A promising solution of this problem seems to be a Built-in Self-Test (BIST) [1, 2]. We propose a test-per-clock BIST method, where the test patterns are fed to the circuit under test (CUT) in parallel, as well as the CUT outputs are being evaluated in a parallel response analyzer, which is mostly a MISR (Multi-Input Shift Register). Usually, the test patterns are produced by a LFSR. Using a plain LFSR satisfactory fault coverage often cannot be reached. Then the LFSR code words must be modified somehow. We propose a method based on a transformation of these code words into deterministic test patterns that are pre-generated by some ATPG tool [3]. The method is an extension of the previously published column matching method [4]. The newly proposed method can take advantage of the don't care (DC) values in the test patterns, which enables us to reach more column matches and significantly reduce the BIST logic.

2 Principles of the Method

The code words of an n-bit LFSR running for p clock cycles can be described by a **C** matrix (code matrix) of dimensions (p, n). They are to be transformed into the test patterns pre-computed by some ATPG tool. The tests are described by a **T** matrix (test matrix). For an r-input CUT and the test consisting of s vectors the **T** matrix will have dimensions (s, r). The tests can be presented either in a form of deterministic patterns or they may contain don't care values, depending on the ATPG algorithm used.

The combinational logic that modifies the **C** matrix vectors to obtain all the **T** matrix vectors will be denoted as an *output decoder*. Since the proposed method is restricted to combinational circuits, the order in which the test patterns are fed to the CUT is insignificant. Finding a transformation from a **C** matrix to a **T** matrix means finding a matching of all the s rows of **T** matrix with any distinct s rows of **C** matrix. The decoder is represented by a Boolean function with n inputs and r outputs, where only values of s terms are defined. This Boolean function can be easily described by a truth table, where the output part corresponds to the **T** matrix and the input part consists of s **C** matrix vectors assigned to the **T** matrix rows. The set of such vectors will be denoted as a *pruned **C** matrix*.

The column-matching method is based on assigning all the **T** matrix rows to some of the **C** matrix rows so that some columns of the **T** matrix will be *equal* to some columns in the pruned **C** matrix. This yields absolutely no logic necessary to implement these **T** matrix columns (output variables of the decoder); they are implemented as mere wires. In [4] a set system based method for putting restrictions for row assignments was proposed. This method might be also used for a column matching exploiting don't cares, however it would be unnecessarily complicated and memory demanding. Thus a more suitable method has been found, namely a method using a *blocking matrix **B***. The blocking matrix is a binary matrix of dimensions (p, s), value "1" in the cell **B**[i, j] indicates that the i-th **C** matrix row may be assigned to the j-th **T** matrix row, "0" value indicates the contrary. At the beginning of the algorithm all the **B** matrix cells are filled with a "1" value, since there are no restrictions for the row assignments. Each of the **T** matrix rows may be assigned to any **C** matrix row. After the p-th **C** matrix column is matched with the q-th **T** matrix column, the **B** matrix cells [i, j] are set to "0" when

$$\mathbf{C}[i, p] \neq \mathbf{T}[j, q] \wedge \mathbf{T}[j, q] \neq \text{don't care}.$$

Thus, rows that contain opposite values in the matched columns cannot be assigned to each other. To assign all the **T** matrix rows to the **C** matrix rows each of the **B** matrix column has to be assigned to some **B** matrix row. The possibility of assigning a row *i* to a column *j* is indicated by a "1" value in **B**[*i*, *j*].

Before each column match is made this assignment must be performed to determine whether it is valid (i.e. whether some assignment can be found). If the assignment fails, the procedure ends. Since most of the test sets including don't cares are not in a compacted form (e.g., there is one test pattern for each of the s-a faults [5]), some test compaction technique [6] should be done after the column matching. Then the matched output variables are removed and the values of the remaining output variables are synthesized by some standard Boolean minimizer [7, 8].

4 Experimental Results - ISCAS Benchmarks

The effectiveness of the method was studied on ISCAS benchmarks [9]. The test sets were generated by ATOM [10], the resulting truth table was minimized by BOOM [7, 11]. In all the cases LFSR running for 5000 cycles was used as a code word generator, the number of the LFSR stages was chosen equal to the CUT primary inputs. The results are show in Table 1. For each circuit the number of its primary inputs is given. The "patterns" column indicate the number of the test patterns, the percentage of the test don't cares is shown too. The "matches" column gives the number of column matches reached and "GEs" indicates the complexity of the output decoder in terms of the gate equivalents.

Table 1. ISCAS benchmarks

	inputs	patterns	% of DCs	matches	GEs
c432	36	277	66.47	25	131
c880	60	559	84.32	52	242
c1355	41	367	18.44	8	900
s208	19	133	70.01	**19**	**0**
s344	24	177	81.59	22	8
s1196	32	734	75.18	24	259

5 Conclusions

A general enhancement of a column matching BIST method was proposed. The column matching is based on transforming the pseudo-random PRPG patterns into deterministic test patterns generated by some ATPG tool. The enhancement consists in the possibility of exploiting the test don't cares which can significantly reduce the amount of the resulting BIST logic. A blocking matrix based method for finding the assignment of the patterns was proposed. The method was tested on ISCAS benchmarks whose complete s-a fault coverage test sets were generated by a tool ATOM.

Acknowledgment

This research was in part supported by grant #102/01/0566 "Built-in Self-Test Equipment Optimization Methods in Integrated Circuits" of the Czech Grant Agency (GACR) and MSM 212300014 , 1999 – 2003.

References

[1] Agarwal, Kime, Saluja: A tutorial on BIST, part 1: Principles. IEEE Design & Test of Computers, vol. 10, No.1 March 1993, pp.73-83, part 2: Applications, No.2 June 1993, pp.69-77

[2] McCluskey, E.J.: BIST techniques. IEEE Design & Test of Computers, vol. 2 No.2 Apr. 1985. pp.21-28, BIST structures. vol. 2 No.2 Apr. 1985. pp. 29-36

[3] Hamzaoglu, I. - Patel, J.H.: New Techniques for Deterministic Test Pattern Generation, Proceedings of the VLSI Test Symposium, pp. 446-452, April 1998

[4] Fiser, P. - Hlavicka, J.: Column-Matching Based BIST Design Method. Proc. 7th IEEE Europian Test Workshop (ETW'02), Corfu (Greece), 26.-29.5.2002, pp. 15-16

[5] Hamzaoglu, I. - Patel, J.H.: New Techniques for Deterministic Test Pattern Generation, Proceedings of the VLSI Test Symposium, pp. 446-452, April 1998

[6] Hamzaoglu, I. - Patel, J.H.: Test Set Compaction Algorithms for Combinational Circuits, Proceedings of the International Conference on Computer-Aided Design (ICCAD), November 1998.

[7] Hlavicka, J. - Fiser, P.: BOOM - a Heuristic Boolean Minimizer, Proc. ICCAD-2001, San Jose, Cal. (USA), 4-8.11.2001, pp. 439-442

[8] Brayton, R.K, et al.: Logic Minimization Algorithms for VLSI Synthesis, Boston, MA, Kluwer Academic Publishers, 1984

[9] Brglez, F. - Fujiwara, H.: A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortan, Proc. of International Symposium on Circuits and Systems, pp. 663-698, 1985

[10] http://www.crhc.uiuc.edu/IGATE/

[11] http://service.felk.cvut.cz/vlsi/prj/BOOM/