Implicit Representations in Test Patterns Compression for Scan-Based Digital Circuits

Jiří Balcárek
Dept. of Computer Science & Engineering
Czech Technical University in Prague, FEL
Prague, Czech Republic
balcaji2@fel.cvut.cz

Petr Fišer, Jan Schmidt
Faculty of Information Technology
Czech Technical University in Prague
Prague, Czech Republic
fiserp@fit.cvut.cz, schmidt@fit.cvut.cz

Abstract— **Current test generation and compression processes still have room for improvement. One important factor is the representation of test vector sets that a tool uses internally. We overview known approaches to this problem, and present our results with implicit representation by instances of the satisfiability problem.**

Keywords: Test compression, testing, implicit representation

I. INTRODUCTION

With the growing complexity of designs, the amount of test patterns need to be stored and applied grows up significantly. That is why the compression of test patterns is needed to decrease the memory requirement and time consumption of the test.

ATPG (Automatic Test Pattern Generator) and compression of test patterns for combinational (or full-scan) circuits is based on manipulation with sets of binary vectors. The sets express the possibilities of choice (at least one vector from the set is needed) or summary (the whole set of vectors is needed). Such sets can be expressed *explicitly,* as an enumeration of vectors, or *implicitly*, by describing the required characteristic properties of the set.

II. PREVIOUS WORK

The most frequently formalism for implicit characterization is the incompletely specified Boolean function. It cannot, however, describe *any* set of vectors. Therefore, while incompletely specified functions can improve size, the description remains partially explicit. Moreover, representations such as Binary Decision Diagrams [28] or And Inverter Graphs [29] are difficult to extend to incompletely specified functions.

There were several test patterns compression methods proposed. Many of these methods are based on using some codes, like statistical codes [3, 6, 7, 8, 10], run-length codes [5, 6, 7, 8, 9], frequency directed codes (FDR) [5] and Golomb codes [11], others are based on XOR networks [12, 13], hybrid patterns [14], folding counters [15, 16], EDT (Embedded Deterministic Test) [26] and reuse of scan chains [17]. Another approaches are based on test pattern compaction [21, 22, 23] and overlapping [1, 18, 19, 20].

III. IMPLICIT REPRESENTATIONS IN TEST COMPRESSION

Our aim is to develop a better compression method for RESPIN [27] and similar architectures. Compression in this case is based on overlapping the test patterns; therefore it requires choosing the right patterns in the right order to ensure optimum compression.

We realized that the set of patterns detecting a fault can be naturally and compactly described by an instance of satisfiability problem (SAT) as it is done in SAT-based ATPG [24, 25]. Furthermore, additional constraints can be easily added to the instance, selecting suitable patterns from the set.

Our tool, SAT Compress [33] uses such constraints to make sure that an overlap with previously generated data is possible. This allows us to generate only suitable vectors, and generate them in the process of compression. The core algorithm is a simple greedy procedure, accepting the first vector that can be overlapped.

A comparison of compressed test lengths in bits for seven different state-of-the-art compression techniques and SAT-Compress are presented in Table I. Even with such a simple method we can obtain similar results such as these state-of-the-art compression methods and in some cases even better. The run time is longer, however, we are aware that test time and not test generation time is the bottleneck.

IV. FUTURE WORK

The existing algorithm and tool harness the basic benefits from the implicit characterization. However, generating, storing and handling the SAT instances can be much improved. The efficiency could then translate into a more thorough heuristic process, giving good results more consistently.

In farther view, RESPIN and the overlapping compression can be replaced with another decompressor, more compatible with SAT description.

V. CONCLUSION

The test vector representation influences substantially the architecture of test compress tools. The advantages of implicit representation by SAT instances are documented on a compression tool for the RESPIN architecture.

ACKNOWLEDGMENT

This research has been supported by M_SMT under research program MSM6840770014 and by the grant of the Czech Grant Agency GA102/09/1668.

REFERENCES

[1] O. Novák, J. Zahrádka, "COMPAS – Compressed Test Pattern Sequencer for Scan Based Circuits," in Proc. of EDCC, 2005, pp. 403-414.

[2] B. Koenemann, "LFSR-Coded Test Patterns for Scan Designs," in Proc. of European Test Conf., IEEE CS Press, 1991, pp. 581-590.

[3] C.V. Krishna, N.A. Touba, "Reducing Test Data Volume Using LFSR Reseeding with Seed Compression," in Proc. of the International Test Conference, 2002, pp. 321-330.

[4] F. Fummi, D. Sciuto, "Implicit test pattern generation constrained to cellular automata embedding," 15th IEEE VLSI Test Symposium (VTS'97), 1997, pp.54-60.

[5] A. Chandra, K. Chakrabarty, "Frequency-Directed Run-Length (FDR) Codes with Application to System-on-a-Chip Test Data compression," in Proc. of VLSI Test Symposium, 2001, pp. 42–47.

[6] P. T. Gonciari, B. M. Al-Hashimi, N. Nicolici, "Variable length input Huffman coding for system-on-a-chip test," IEEE Trans. Computer-Aided Design, vol. 22, 2003, pp. 783–796.

[7] H. Ichihara, A. Ogawa, T. Inoue, and A. Tamura, "Dynamic test compression using statistical coding," in Proc. ATS, 2001, pp. 143–148.

[8] H. Ichihara, K. Kinoshita, I. Pomeranz, and S. Reddy, "Test transformation to improve compaction by statistical encoding," in Proc. Int. Conf. VLSI Design, 2000, pp. 294–299.

[9] A. Jas and N. A. Touba, "Test vector decompression via cyclical scan chains and its application to testing core-based designs," in Proc. ITC, 1998, pp. 458–464.

[10] A. Jas, J. Ghosh-Dastidar, and N. A. Touba, "Scan vector compression/decompression using statistical coding," in Proc. VLSI Test Symp., 1999, pp. 114–120.

[11] A. Chandra, K. Chakrabarty, "Test Data Compression for System-on-a-Chip Using Golomb Codes," 18th IEEE VLSI Test Symposium (VTS'00), 2000, pp.113-121.

[12] I. Bayraktaroglu and A. Orailoglu, "Test volume and application time reduction through scan chain concealment," in Proc. Design Automation Conf., 2001, p. 151–155.

[13] S. M. Reddy, K. Miyase, S. Kalihara, and I. Pomeranz, "On test data volume reduction for multiple scan chain design," in Proc. VLSI Test Symp., 2002, pp. 103–108.

[14] D. Das and N. A. Touba, "Reducing test data volume using external/ LBIST hybrid test patterns," in Proc. ITC, 2000, pp. 115–122.

[15] S. Hellebrand, H.-G. Liang, and H.-J.Wunderlich, "Amixed mode BIST scheme based on reseeding of folding counters," in Proc. ITC, 2000, pp. 778–784.

[16] H.-G. Liang, S. Hellebrand, and H.-J. Wunderlich, "Two-dimensional test data compression for scan-based deterministic BIST," in Proc. ITC, 2001, pp. 894–902.

[17] R. Dorsch and H.-J. Wunderlich, "Tailoring ATPG for embedded testing," in Proc. ITC, 2001, pp. 530–537.

[18] C. Dufaza, H. Viallon, C. Chevalier, "BIST hardware generator for mixed test scheme," edtc, European Design and Test Conference (ED&TC '95), 1995, pp. 424-431.

[19] C. Su, K. Hwang, "A Serial Scan Test Vector Compression Methodology," in Proc. ITC, 1993, pp. 981-988.

[20] Daen, W., Mucha, J,: Hardware Test Pattern Generation for Built-in Testing. Proc. of IEEE Test Conference, 1981, pp. 110-113.

[21] Seiji Kajihara, Kewal K. Saluja, "On Test Pattern Compaction Using Random Pattern Fault Simulation," vlsid, Eleventh International Conference on VLSI Design: VLSI for Signal Processing, 1998, pp.464-470.

[22] A. Giani, Shuo Sheng, M. Hsiao, V.D. Agrawal, "Compaction-based test generation using state and fault information," ats, , Ninth Asian Test Symposium (ATS'00), 2000, pp.159-165.

[23] I. Pomeranz, S. M. Reddy, "A New Approach to Test Generation and Test Compaction for Scan Circuits," Proc. of the conference on Design, Automation and Test in Europe (DATE'03), 2003, pp. 11000-11006.

[24] T. Larrabee, "Test Pattern Generation Using Boolean Satisfiability," IEEE Transactions on Computer-Aided Design, 1992, pp. 4-15.

[25] Drechsler, R., Eggersglüß, S., Fey, G., Tille, D., "Test Pattern Generation using Boolean Proof Engines," Publisher Springer Netherlands, ISBN 978-90-481-2360-5, 2009, XII, p. 192.

[26] Rajski, J., "Embedded Deterministic Test" IEEE Trans. on CAD, vol. 23, No. 5, 2004, pp. 776-792.

[27] Schafer L., Dorsch R., Wunderlich H.J., "RESPIN++- Deterministic Embedded Test," Proc. of the European Test Workshop, 2002, pp.37-42.

[28] R. Drechsler, D. Sieling, "Binary decision diagrams in theory and practice," Springer Trans., Berlin, 2001, pp. 112-136.

[29] A. Mishchenko, S. Chatterjee, R. K. Brayton, "DAG-aware AIG rewriting: a fresh look at combinational logic synthesis". In 43th Annual ACM IEEE Design Automation Conference, San Francisco, CA, USA, 2006, pp. 532-535.

[30] A. Chandra, K. Chakrabarty, "Test Data Compression and Test Resource Partitioning for System-on-a-Chip Using Frequency-Directed Run-Length (FDR) Codes," IEEE Transactions on Computers, vol. 52, May 2003, to appear.

[31] I. Hamzaoglu, J.H. Patel, "Reducing Test Application Time for Full Scan Embedded Cores," Proc. Of Int. Symp. on Fault Tolerant Computing, 1999, pp. 260-267.

[32] Janak H. Patel, Ilker Hamzaoglu, "Test Set Compaction Algorithms for Combinational Circuits," iccad, International Conference on Computer-Aided Design (ICCAD '98), 1998, pp.283-289.

[33] J. Balcárek, "Test Patterns Compression Techniques Based on SAT Solving for Scan-Based Digital Circuits," Počítačové architektury&diagnostika, Soláň (CR), 2009, pp. 26-31.

TABLE I. COMPARISON OF THE TEST DATA SIZE FOR DIFFERENT COMPRESSION TECHNIQUES

Bench. circuits	MinTest [32]	Stat. Coding [10]	LFSR Reseeding [3]	Illinois Scan [28, 31]	FDR Codes [5, 30]	EDT [26]	RESPIN++ [27]	COMPAS [1]	SAT-Compress
s5378	20,758	15,417	6,180	14,572	12,346	-	17,332	2,148	2,407
s9234	25,935	19,912	12,112	27,111	22,152	-	17,198	11,594	9,928
s13207	163,100	52,741	11,285	109,772	30,880	10,585	26,004	4,163	10,457
s15850	58,656	49,163	12,438	32,758	26,000	9,805	32,226	8,234	12,987
s35932	21,156	-	-	-	22,744	-	-	1,860	5,096
s38417	113,152	172,216	34,767	96,269	93,466	31,458	89,132	24,198	19,291
s38584	161,040	128,046	29,397	96,056	77,812	18,568	63,232	7,291	14,271