International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 18 (2015) pp 39853-39857
© Research India Publications. http://www.ripublication.com

Test Data Compression Techniques Emphasizing Linear Decompressor Based Schemes: A Review

S. Rooban,

[1]Research Scholar, Sri Krishna Institute of Technology, Chennai, India skit.ecsrn@gmail.com

Dr. R. Manimegalai

Professor (CSE), Park college of Engineering and Technology, Coimbatore, India mmegalai@yahoo.com

Abstract

One of the increasingly difficult challenges in testing a chip is test data volume. This large volume of test vector must be stored on the tester and then transferred to the inputs of the core. Shrinking process technologies increases the number of tests and the corresponding test data volume and test time. Researchers have proposed numerous test compression techniques that reduce the amount of test data and test time. Among them, the linear decompression based schemes have gained popular for its simple control logic and very high encoding flexibility. This work presents a comprehensive survey of linear decompression based schemes and discusses the various types in that test data compression schemes. It gives the framework for understanding this compression technique and provides a platform for further development in this area.

Keywords: Automatic Test Pattern Generation (ATPG), Linear Feedback Shift Register (ATPG), Circuit Under Test (CUT), System on Chip, Automatic Test Equipment (ATE)

Introduction

Advancement in nanometre IC fabrication CMOS technology introduces scaling of transistor dimensions and allows a large number of devices to be fabricated on a single chip. Continuous increase in chip complexity needs excessive test data volume even for single stuck-at-faults with single-detection. This highly growing test data volume for SoC test is one of the major challenges in today's circuit design. The ATE storage requirement depends on the amount of test data. Total test time required for testing a chip also depends on the test data volume as well as the test data bandwidth for transferring the data [1].Testing cannot be done any faster than the amount of time required to transfer the test data[2]. Test data bandwidth between the tester and the chipis relatively low and limits the testing speed.

Approaches to Increase the Testing Speed

The three approaches [2] to increase the testing speed to overcome the limits of test data bandwidth are Stand-alone BIST, Hybrid BIST and Test data compression. Table 1 shows the comparison of the three approaches. From the Table 1 we have observed that Test data compression approach provides more fault coverage with less test pattern than other two approaches. As the amount of test data is increased, test compression has become very attractive and the additional hardware overhead is relatively low and significant reduction (10x or even 100x) in the amount of test data that must be stored on the tester can be achieved. One benefit of test data compression is that it can extend the life older "legacy" testers that may have limited memory by making it possible to fit all of the test data in the tester memory. Even for testers that have plenty of memory, test compression is still very attractive because it can reduce the test time for a given test data bandwidth [3].

Table 1: Comparison of Approaches to Increase Testing Speed

Stand-alone BIST	Hybrid BIST	Test data compression
1. Eliminate the need for tester storage.	Stores part of test data on the tester to increase fault coverage.	Requires additional hardware before and after the scan chain.
2. Useful for performing self-test in the field avoids the tester requirements.	Need to apply more test pattern to CUT than test data compression.	Easier to adopt in industry because it's compatible with the conventional design rules.
3. Fault coverage is low.	Requires longer test application time.	It generates the complete set of optimizable test pattern with respect to the desired fault coverage by ATPG.

Test Data Compression

The idea in test data compression is to compress the amount of test data both stimulus and response that is stored on the tester. A generic architecture for several compression techniques used to reduce the data volume and the tester channel requirements is shown in Fig. 1. (A - Decompressor, B - Compactor)

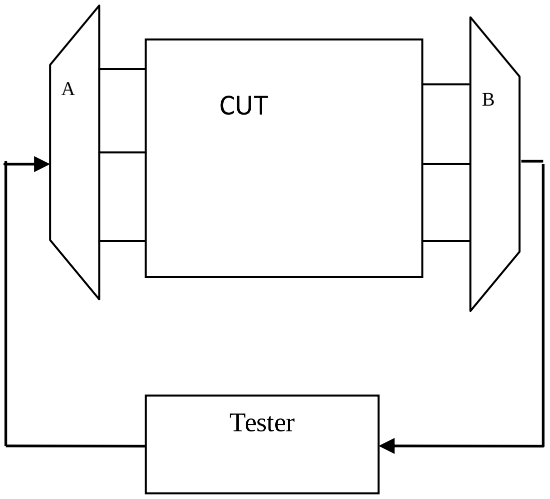

Fig 1: Generic Architecture for Test Compression

The decompressor takes the compressed test vector as input from the tester, expands them and sends the expanded test vectors to the CUT. The output compactor converts long output response sequences into short signature. This additional on-chip hardware allows the test data to be stored in compressed form. Depending on theimplementation of the decompression hardware, the schemes for test data compressions are classified in to three schemes namely Code-based schemes, Broadcast-scan-based schemes and Linear-decompression-based schemes. Table 2 shows the characteristics of the three test data compression schemes. The Linear decompression based schemes typically achieve better encoding efficiency than other two types, and they are the dominant approach used in the industry today [4].

Linear Decompression-Based Scheme

Any decompressor that consists of only XOR gates and flip-flops is a linear decompressor. Linear decompressor that expands an m-bit compressed stimulus from the tester into an n-bit (n>m) stimulus, if there exists a Boolean matrix $\mathbf{A}_{\mathbf{nxm}}$ such that the set of test vectors that can be generated by the linear decompressor is spanned by **A**. A test vector Z can be compressed by a particular linear decompressor if and only if there exists a solution to a system of linear equations, **A**X = Z, where **A** is the characteristic matrix of the linear decompressor and X is a set of free variables stored on the tester[3]. The number of free variables determines the encoding capacity achievable by the linear decompressor. A figure of merit for linear decompressors is encoding efficiency. It’s defined as (Specified Bits in Test Set ÷ Bits Stored on Tester). How close linear decompressor encoding efficiency is to 1 is a measure of its optimality.

Table 2 Characteristics of different test data compression schemes

Schemes	**Code-based scheme** (use data compression codes to encode the test cube)	**Broadcast-scan-based scheme** (Broad-casting the same value to multiple scan chains)		**Linear-decompression-based scheme** (use only linear operations using LFSR and exclusive-OR networks)	
		Static reconfiguration	Dynamic reconfiguration	Combinational	Sequential
Pros	Exploits correlation in specified bits	Simple decompressor	More encoding flexibility than static	Uses only combinational gates	Low linear dependency
	Usable on any set of test cubes	Efficient one-step ATPG	One-step ATPG	Very simple control logic	Very high encoding flexibility
Cons	Not efficient in exploiting don’t-cares	High linear dependency	More control than static	Two-step ATPG	Two-step ATPG
	More complex control logic	Lower encoding flexibility	Less encoding flexibility than sequential linear decompressor	Encode each slice independently	More complex decompressor

A. Combinational Linear Decompressor

The combinational linear decompressor implements the characteristic matrix by using XOR network [5][6] and its encoding capability at a shift cycle is restricted to the number of inputs of the XOR network. Tester provides XOR-ed subset of channels to each scan chain [7] [8].The advantages compared with sequential linear decompressors are simple hardware and control. One approach for improving the encoding efficiency of combinational linear decompressor [4] is to dynamically adjust the number of scan chains that are loaded in each clock cycle. If four clock cycles are used to load the highly specified bits to the scan chain, only 25% of the scan chain is loaded in each clock cycle. In [9] to test CUT with a single scan chain, first the compressed neighbouring bitwise exclusive-or (NB-XOR)-ed difference vectors from the ATE are transferred to an on-chip decoder which decompresses them into the original NB-XORed difference vectors. And then the decompressed vectors are fed into the inverse NB-XOR block, which consists of a simple pair of one XOR gate and one flip-flop, to transform them into the original low-power scan vectors. Finally the transformed vectors are shifted into the internal scan chain. It requires less area overhead than the CSR (Cyclical Shift Register) and it does not require any knowledge or modification of the internal structure of the core.

B. Sequential Linear Decompressor

Sequential linear decompressors are formed by linear finite-state machines (LFSRs, Cellular automata, or ring generators) [10]. Sequential linear decompressor encodes the scan slices by using free variables from the previous clock cycle to encode the scan slice in the current clock cycle [2].This provides greater flexibility and advantages over combinational decompressors. Flexibility and encoding efficiency can be increased by adding more number of flip-flops. Well-designed sequential decompressor with sufficient number of flip-flops gives greater than 0.95 encoding efficiency [11] [12]. The sequential linear decompressor inserts a linear finite-state machine such as LFSR or ring generator between the decompressor inputs and the XOR network [13]-[16]. It improves the encoding capability by utilizing the compressed test stimuli shifted-in at both current and previous cycles to encode the test stimulus needed at the current shift cycle. In [17] T_D is the un-compacted test set and T_{diff} is the corresponding difference vector test sequence. The advantage of using T_D for compression is that the decompression architecture for on-chip pattern generation does not require a separate CSR and reduces the hardware overhead.

C. Fixed Length Sequential Linear Decompressor

We achieve compression by storing only the seed in the tester instead of full test cubes. Combination of LFSR and XOR networks expands the seed from the tester and loads the scan chain with the help of additional free variables. Simple sequential linear decompression is performed by just shift the same number of free variables for every test cube. Control logic is not complex but limited encoding efficiency for the worst-case most heavily specified test cube. In this fixed length or static reseeding approaches [14]-[15], all the specified bits in a test cube are encoded by using an LFSR seed, and the LFSR size has to be no less than the number of specific bits in the test cube. Increasing LFSRs size as the increasing number of specified bits in the test cube can be controlled only by decompress a limited number of scan slices per seed [13] [15]. In [18] LFSR reseeding is used to generate update flag, hold flag and test data for each test cube. The last value shifted into the scan chain is repeatedly shifted into the scan chain and the data from the LFSR is ignored decided by the value of hold flag. It reduces the transitions and increases the compression with the hardware overhead for the hold flag. Idle time of the tester when the LFSR is running in autonomous mode is eliminated by the shadow register. It holds the data coming from the tester when the LFSR is busy [19] [20].

D. Variable Length Sequential Linear Decompressor

Variable length sequential linear decompressor improves its encoding efficiency by varying the number of free variables. It uses only as many free variables as are needed to encode each test cube. In Heavily specified scan slice, scan shifting can be stopped by the extra gating channel for one or more clock cycles and it allows the LFSR to accumulates sufficient number of free variables to solve the current scan slice before proceeding to the next one[16] [20]. Reduction in compression caused by the extra gating channel can be avoided by the idea of initialize the counter with the value specified by the first b-bits coming from the tester during the first clock cycle [21]. These b-bits denote how many clock cycles should be used for decompressing the test cube. As the encoding efficiency increases with the number of flip-flops used, one method to achieve more number of flip-flops without incurring a lot of overhead is to configure the scan chains themselves into a large LFSR as originally suggested in [22]. Applying the compressed test stimuli continuously during shift, the dynamic reseeding approaches increase the encoding capability significantly while allowing the use of a smaller LFSR [16][23]. In [24] by dividing the VIHC (Variable-length Input Huffman Coding) decompressor into Huff-decoder and CGU (Control and Generator Unit) allows Huff-decoder to continue loading the next code word while the CGU generates the current pattern. Thus the Huff-decoder is interrupted only if necessary and leads to reduction in TAT (Test Application Time). An additional advantage of this on-chip parallel decompressor is that having the two components working in two different clock domains, this facilitates data delivery to the CUT at the on-chip test frequency using only one ATE (Automatic Test Equipment) channel therefore better resource utilization. In [25] decompression is achieved by for each scan slice, first all the scan cells are assigned to the same values as the previous scan contents, then the corresponding bits are flipped one by one in the following cycles according to the configuration vectors. If the number of bits to be inverted is small, due to the low filling rate in test set, co-joint reduction in test data volume and power consumption is achieved.

E. Combined Linear and Non-Linear Decompressor

Combining the linear and nonlinear decompressor provides higher compression by overcoming all the limitations when they are operating separately. Selecting the solution to the system of linear equations for each test cube is in such a way that they can be effectively compressed by the statistical code [15]. Statistical code reduces the number of bits stored on the tester for the linear decompressor. Combining the linear decompressor with dictionary coding, linear decompressor generates the code for the scan slice if it’s not available in the dictionary [26]. Nonlinear decompressor identifies scan chains that are compatible with a nonlinear combination of two other scan chains which is placed between linear decompressor and the scan chains [27] [28]. Compatible relationship among the scan chains is difficult to identify from the increasing scan length and the increased number of test patterns.

Conclusion

Selecting an effective Test data compression from the number and diversity of available techniques is hard and the ultimate selection depends on several parameters. The parameters are implementation context, fault coverage, testing time, area overhead and the effect on the design flow. In this survey paper we have tried to cover the test data compression techniques proposed by researchers in linear decompression based schemes. This analysis shows various categories in linear decompression and found that the linear sequential decompressor has better encoding efficiency.

References

[1] Anushuman Chandra, KrishnenduChakrabarty, ; “Combining Low-Power Scan Testing and Test Data Compression for System-on-a-Chip”, Proceedings 38th Design Automation Conference, DAC-2001, 18-21 June, Newyork, p. 166-169.

[2] Nur A. Touba., 2006, “Survey of Test Vector Compression Techniques” IEEE Design & Test of Computers, Vol. 23, No. 4, 2006, p. 294-303.

[3] Laung-Terng,, Wang Cheng-Wen, Wu Xiaoqing Wen., “VLSI Test Principles and Architectures”, Elsevier, 2006.

[4] Xijiang Lin, Mark Kassab, JanuszRajski.; “Using Dynamic shift to Reduce Test Data Volume in High Compression Designs”, 19th IEEE European Test Symposium 2014(ETS),, May 26-30, Germany.

[5] B. Keller et al.; “OPMISR: The Foundation for Compressed ATPG Vector”, Proceedings IEEE International Test Conference, 2001, 30th Oct-01 Nov, Maryland, p. 748-757.

[6] C. Barnhart et al, “Extending OPMISR Beyond 10X Scan Test Efficiency”, IEEE Design & Test of Computers, Vol. 19, No. 5, Sep/Oct 2002, p. 65-73.

[7] Bayraktarogl, Orailoglu, “Concurrent Application of Compaction and Compression for Test Time and Data volume Reduction in Scan designs”, IEEE Transaction on Computers., Vol. 52, No. 11, 2003, p.1480-1489.

[8] Koenemann, Barnhart., “Real-Time Decoder for Scan Test Patterns”, U.S. Patent No. 6, 611, 933 August 2003.

[9] Jaehoon Song, Hyunbean Yi, Doochan Hwang, Sungju Park.; “A Compression Improvement Technique for Low-Power Scan Test Data”, IEEE Region 10 Conference, TENCON-2006, 14-17 Nov, Hong Kong.

[10] Mrugalski, Rajski., 2004 “Ring generators: New devices for embedded test applications, IEEE Transaction on Computer Aided Design of Integrated Circuits & Systems, Vol. 23, No. 9, 2004, p. 1306-1320.

[11] Krishna, Jas, Touba, “Test Vector Encoding using Partial LFSR Reseeding”, Proceedings IEEE International Test Conference 2001, 30th Oct-01 Nov, Maryland p.885-893.

[12] Rajski, Tyszer, Kassab., “Embedded Deterministic Test”, IEEE Transaction on Computer Aided Designs., Vol. 23, No. 5, 2004, p.776-792.

[13] P. Wohl et al., “Efficient Compression of Deterministic Patterns into Multiple PRPG Seeds”, Proceedings IEEE International Test Conference 2005, 8th Nov, Texas, p. 925-935.

[14] S. Hellebrand et al., “Built-in Test for Circuits with Scan Based on Reseeding of Multiple-Polynomial Linear Feedback Shift Register”, IEEE Transactions on Computers, Vol. 44, No. 2, Feb 1995, p.223-233.

[15] C. V. Krishna et al., “Reducing Test data Volume Using LFSR Reseeding with Seed Compression”, Proceedings IEEE International Test Conference 2002, 10th Oct, p. 321-330.

[16] B. Koenemann et al., 2001, “A Smart BIST Variant with Guaranteed Encoding”, Proceedings 10th Asian Test Symposium 2001,. ATS-2001, 19th -21 Nov, Kyoto, p. 325-330.

[17] Anushuman Chandra, KrishnenduChakrabarty., 2003, “Test Data Compression and Test Resource Partitioning for System-on-a-Chip Using Frequency-Directed Run-Length (FDR) Codes”, IEEE Transaction on Computers, Vol. 52, No.8, 2003, p. 1076-1088.

[18] Jinkyu Lee and Nur A. Touba., 2004, “Low Power Test Data Compression Based on LFSR Reseeding”, Proceedings IEEE International Conference on Computer Design 2004(ICCD-04), 11-13th Oct, p. 180-185.

[19] P.Wohl et al., “Efficient Compression and Application of Deterministic Patterns in a Logic BIST Architecture”, Proceedings 41st Design Automation Conference 2003 (DAC 03), 2-6th June, p. 566-569.

[20] E. H. Volkerink, S. Mitra, “Efficient Seed Utilization for Reseeding Based Compression”, Proceedings 21st VLSI Test Symposium 2003 (VTS 03), 27th April-1st May, p. 232-237.

[21] C. V. Krishna, N. A. Touba, “3-Stage Variable Length Continuous-flow Scan Vector Decompression scheme”, Proceeding VLSI Test Symposium 2004, 25-29th April, p. 79-86.

[22] J. Rajski, J. Tyszer., “Test Data Decompression for Multiple Scan designs with Boundary Scan, IEEE Transaction on Computers, Vol. 47, No.11, 1998, p. 1188-1200.

[23] J. Rajski et al., May 2004, “Embedded Deterministic Test”, IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems, Vol. 23, No. 5, 2004, p. 776-792.

[24] Paul T. Gonciari, Bashir M. Al-Hashini, and Nicola Nicolici., June 2003, “Variable-Length Input Huffman Coding for System-on-a-Chip Test”, IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems, Vol. 22, No. 6, 2003, p. 783-796.

[25] Youhua Shi, Nozomu., 2005, “Low Power Test Compression technique for Designs with multiple

Scan Chains" IEEE Proceedings of the 14th Asian Test Symposium 2005, 18th-25Dec, p. 386-389.

[26] X. Sun, L. Kinney., 2004, "Combining Dictionary Coding and LFSR Reseeding for Test Data Compression", Proceeding IEEE 41st Design Automation Conference 2004, 7th -11th July, California, p. 944-947.

[27] L. Li, K. Chakrabarty., "Efficient Space/time Compression to reduce test data volume and testing time for IP cores", Proceeding 18th VLSI Design Conference 2005, 3-5th Jan, Kolkatta, p. 53-58.

[28] S. Ward, C. Schattauer., "Using Statistical Transformation to Improve Compression for Linear decompressors, Proceeding 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems 2005, 3-5th October, California, pp. 43-50.