

STAGE: A Decoding Engine Suitable for Multi-Compressed Test Data

Bernd Koenemann
Cadence Design Systems, Inc.,
San Jose, CA
e-mail: berndk@cadence.com

Abstract:

Most of the recently discussed test stimulus data compression techniques are based on the low care bit densities found in typical scan test vectors. Data reduction primarily is achieved by compressing the don't-care bit information, while maintaining the care bit data. The original care bit density, hence, dominates the theoretical compression limits.

This paper discusses potential on-chip hardware decoder architectures that allow for combining care bit oriented methods with test cube clustering to achieve multi-level test stimulus compression that reduces the data for both care bits and don't-care bits.

Introduction and Background

Test data compression using simple on-chip decoders has become a very hot topic for research as well as in commercial scan test generation tools. Most of the work takes advantage of the fact that scan test vectors from Automatic Test Pattern Generation (ATPG) contain relatively few care bits which must be at specific logic values for target fault detection. The remaining bit values in the test vectors are don't-cares that are filled in by arbitrary fill algorithms. How these properties can be exploited for test data compression by intelligent re-seeding of on-chip Linear Feedback Shift Registers (LFSRs) was first pointed out in [1]. Under certain statistical assumptions the average size of the LFSR seed vectors needed to represent the care bit values is only slightly larger than the number of care bits. Thus, if the original care bit density in a test vector is 2%, then a close to 50x stimulus data compression ratio should be achievable.

Subsequent technical papers introduced refinements of the original LFSR-Coding method and experimentally confirmed that the theoretical compression limit can indeed be approached [e.g., 2, 3, 4].

The concept can be generalized to work with decoding circuits other than LFSRs. In fact, any combinational or sequential linear network can be used. Even the simplest form, broadcast scan, which uses simple fan-out from each scan-in pin to several parallel scan chains, produces very good compression results [5, 6, 7].

On-chip decoding can be complemented by tester-resident methods that algorithmically generate the fill data for the don't-care bits on the tester. A Run Length Encoding (RLE) approach described in [7, 8] uses the repeat features available on certain testers. Experimental data [7] suggest that the tester-resident RLE method can be combined with moderate on-chip broadcast scan (e.g., 10x or 20x fan-out) for very dramatic data volume reductions (e.g., 100x).

Weighted Random Patterns

In addition to low care bit densities there exist other useful scan test properties. Specifically, test cubes (care bits in a test vector) for multiple faults tend to form clusters of cubes that differ from each other in only very few bit position. For example, the stuck-at fault test cubes for any multi-input AND gate differ from each other in at most 2 bit positions irrespective of the width of the AND gate. Each test cube cluster can be represented by a common base cube and a sparse, highly compressible, difference vector for each unique test cube within the respective cluster.

Weighted Random Pattern (WRP) testing encodes the base cube information into multi-bit weight values for each scan cell in the circuit under test [9, 10]. The weight values determine the strength and direction for biasing output data from a Pseudo Random Pattern Generator (PRPG) towards the base cube values. Industrial WRP versions tend to use ATPG to "find" test cube clusters and derive suitable weight value sets for each such cluster. Then, groups of "trial" vectors with different PRPG seeds are fault-simulated to determine which seeds actually produce useful tests that detect new faults. Extensive experience with WRP has shown that the combined storage for the weight value sets and effective PRPG seeds can be 5x to 20x less than the storage required for traditional stored pattern test set with equivalent fault coverage. WRP has been and still is in production use for some of the industry's most complex chips [12, 13].

Several proposals for on-chip WRP implementations have yielded have been pursued [e.g., 17, 18]. However, the multi-bit nature of the weight values and the resulting size of each weight value have remained obstacles for

finding a suitable partitioning between tester-resident and on-chip features.

The original WRP implementation, for example, uses 4-bit weight values for each scan cell. Each weight value set, thus, requires 4 times more data than the stimulus data portion of a normal test vector. The test data reduction comes from looping over the same weight value set in tester memory multiple times with different PRPG seeds to derive several test vectors. Given that the weight logic used for biasing the PRPG output values is very small and suitable for on-chip implementation, it may be tempting to put the PRPG and weight logic on-chip and supply the weight values from the tester. However, that approach creates a bandwidth problem: instead of having to deliver a single test vector bit to each scan-in pin for each scan cycle, 4 weight value bits would have to be delivered from the tester.

In the following it will be investigated how a combination of reducing the weight value bit-width and compressing the information in each weight value set could help overcome the bandwidth problem and even make it possible to store compressed weight value sets on-chip.

Hybrid and Biased Random Patterns

Hybrid Random Pattern (HRP) testing [14] reduces each weight value to 2 bits and simplifies the weight logic. Figure 1 illustrates a possible HRP weight logic implementation for a single scan chain.

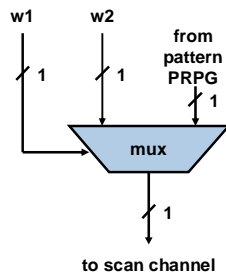


Figure 1: Weight Logic for HRP

One of the 2 weight value bits, w_1 , selects between a pattern PRPG and the second weight value bit, w_2 , as the source of data for the scan chain input. If the latter is selected, then the value of w_2 determines whether a 0 or a 1 is scanned into the scan chain for the respective scan cycle.

Biased Random Patterns (BRP) testing tries to go one step further by using single-bit weight values that modulate the polarity of a uniformly weighted pseudo-random pattern stream towards the respective base cube values [15]. A possible implementation of the BRP weighting logic for a single scan chain is shown in Figure 2.

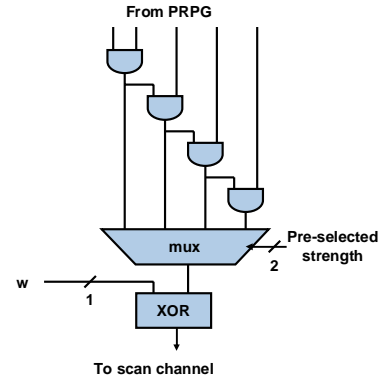


Figure 2: Weight Logic for BRP

The logic shown in Figure 2 allows for pre-selecting a uniform 0-probability of 1/4, 7/8, 15/16, or 31/32. This pre-selected probability normally is not changed within a scan load operation. A single, cycle-by-cycle weight value bit w selectively inverts the uniformly weighted data such that certain scan cells will receive predominantly 1s while others receive predominantly 0s. The strength of the uniform weighting determines how similar the resulting vectors will be to the base vector representing a cluster. Stronger weighting effectively reduces the effective Hamming distance between the generated test vectors while improving the ability to match wider cubes. Weaker weighting increases the variety of the generated vectors but makes it harder to match larger test cubes.

The practical value of using different pre-selected 0-probabilities was seen in a relatively recent application of a simple on-chip BRP implementation in an industrial high-end custom processor product [16]. The overall concept and effectiveness of biased patterns also has been investigated in earlier experimental work [17].

Combining Clustering Effects and Care Bits

The test compaction algorithms for HRP and BRP allow for some limited number of conflicts between merged test cubes but otherwise work very much like normal merging for stored pattern tests [19]. Although this tolerance of conflicts enables somewhat better merging, the overwhelming majority of scan cells in large designs still must be expected to remain don't cares. In other words, only the weights for a small subset of scan cell positions must be at specific values, while the weight values for the remaining majority of scan cells could be assigned arbitrarily without affecting the target fault coverage probabilities. Hence, the concept of care bit and don't-care bit positions carries over to weight value sets. That makes it possible to apply sparse care bit techniques similar to those used for test vector compression, for example PRPG re-seeding. The idea is to encode the information repre-

senting the weight values for the care bit positions into a PRPG seed and then cycle the PRPG to derive the weight values. The challenge for an on-chip implementation is to find a PRPG architecture that allows for economically storing at least one compressed weight value set on-chip.

Figure 3 illustrates a possible PRPG architecture that uses dense RAM/ROM instead of flip-flops for storing and decompressing weight value sets on-chip.

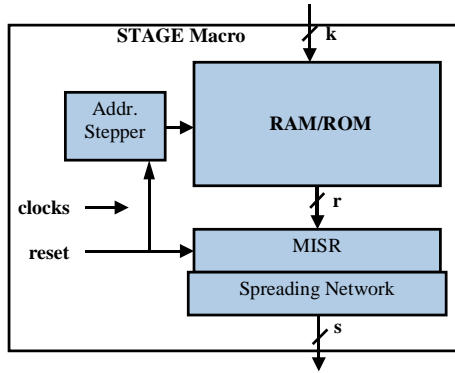


Figure 3: Generic Decompression Architecture

The so called Store And Generate (STAGE) macro consists of a dense RAM or ROM for storing at least one compressed weight value set, an address stepper, a Multiple-Input Signature Register (MISR) and a spreading network (Exclusive OR network). For each scan load, the MISR and address stepper are reset and then cycled along with the scan chains. The address stepper steps through some RAM/ROM address space containing the compressed weight value set information. Because the MISR and spreading network are linear circuits, the resulting output value from the STAGE macro for each scan cycle is a predictable Exclusive-OR sum of the RAM/ROM contents. The outputs of the STAGE macro, rather than feeding the scan chains directly, drive the weight value inputs of on-chip weight logic like that shown in Figures 1 or 2.

Monte Carlo simulations of various STAGE macro configurations (different memory widths and depths) have confirmed that the probability of finding linear dependencies in the output streams matches the theoretical probability for Pseudo-Random Patterns. The STAGE macro, hence, performs like a large. The dense RAM/ROM technology allows for economically storing much larger seed words than would be affordable with conventional flip-flop based PRPGs.

The encoding of weight value sets into STAGE seeds works exactly like the encoding of test vectors into LFSR seeds. Each output value generated by STAGE is an Exclusive OR sum of seed values in the RAM/ROM. If a particular STAGE output value is associated with a care bit scan cell position that requires a specific weight value, then a linear equation is created for that output. Don't-

care scan cell positions, by contrast, create no equation. It should be noted that for multi-bit weighting schemes more than one equation may be generated for any particular care bit position. That is illustrated in Figure 4 for the case of HRP.

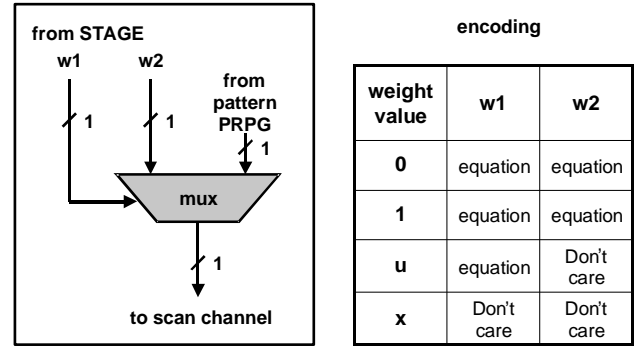


Figure 4: Encoding Equations for HRP

The value 'u' in the encoding table represents a care bit position that requires a pseudo-random value instead of a fixed 0 or 1. An 'x' denotes a don't-care. If a 'u' is required, then w1 must be set to select the pattern PRPG as the data source, while w2 can be left arbitrary. Hence the required value for w1 creates an equation, but no equation is needed for w2. If a fixed value is needed for the scan cell position, then w1 must select w2 as the data source and w2 must provide the correct value, meaning that both w1 and w2 create an equation.

Estimate of On-Chip Memory Overhead

Since HRP, unlike conventional stored patterns, allows for some conflicting values during the merging of test cubes, it must be expected that the resultant care-bit density in the weight value sets is higher than in normal test vectors. Assuming a 50% increase a design with 2% test vector care bit density would have a corresponding HRP weight value care bit density of 3%. Anticipating 2 bits of information required for a HRP care bit weight value, encoding a weight value set for all care bits requires the equivalent of roughly 6% of the scan cells on the chip. Assuming that SRAM storage is 4 times as dense as flip-flops, the area of a STAGE SRAM sufficient to hold one encoded weight set would be equivalent to the area of 1.5% of all scan cells on the chip. It is reasonable to assume that flip-flops can account for 30% of the area in logic-dominated chips, meaning that the 1.5% of scan cells roughly translates into 0.5% overall area.

The overhead for a BRP approach would be less, but more work is needed to better understand how efficient BRP can be in terms of test vector count and test time.

Summary and Future Work

This paper introduced a particular hardware design suitable for decoding test data that are doubly compressed utilizing low care bit density and test cube clustering effects. Initial rough estimates of the overhead for an on-chip implementation are encouraging enough for more investigation.

The next steps are to further explore different partitioning options, particularly refinements of the interface between tester-resident and on-chip resources, and to perform HRP/BRP test generation experiments to validate the sizing estimates.

Acknowledgement

The work presented in this paper was performed at International Business Machines Corporation.

References

1. B. Koenemann, "LFSR-Coded Test Patterns for Scan Designs", ETC '91, pp.237-242, 1991
2. S. Vankataraman et al. "An Efficient Bist Scheme Based On Reseeding Of Multiple Polynomial Linear Feedback Shift Registers", IC-CAD '93. pp. 572-577, 1993
3. Pieter M. Trouborst, "LFSR Reseeding as a Component of Board Level Test", ITC '96, pp. 58-96, 1996
4. N. Zacharia et al. "Two-Dimensional Test Data Decompressor for Multiple Scan Designs, ITC '96, pp. 186-194, 1996
5. K. Lee, J. Chen, and C. Huang, "Using a Single Input to Support Multiple Scan Chains," Proc. Int'l Conf. Computer-Aided Design (ICCAD 98), ACM Press, New York, 1998, pp. 74-78.
6. F. Hsu, K. Butler, and J. Patel, "A Case Study on the Implementation of the Illinois Scan Architecture," Proc. Int'l Test Conf. (ITC 01), IEEE Press, Piscataway, N.J., 2001, pp. 538-547
7. C. Barnhart et al., "Extending OPMISR beyond 10x Scan Test Efficiency", IEEE Design & Test Magazine, Sept.-Oct. 2002, pp. 65-73
8. C. Barnhart et al., "OPMISR: The Foundation for Compressed ATPG Vectors," Proc. Int'l Test Conf. (ITC 01), IEEE Press, Piscataway, N.J., 2001, pp. 748-757
9. J.A.Waicukaski et al., "A Method for Generating Weighted Random Test Patterns", IBM Journal of R&D, vol. 3 no.2, 1989, pp. 149-161
10. R. Kapur et al., "Design of an Efficient Weighted Random Pattern Generation System", Proc. International Test Conference, 1994, pp. 491-500
11. B. Koenemann, "A Pattern Skipping Method for Weighted Random Pattern Testing", Proc. European Test Conference, 1993, pp. 418-425
12. P. Gillis et al., "Test Methodologies and Design Automation for IBM ASICs", IBM Journal of R&D, vol. 40 no. 4, 1996, pp. 461-474
13. T. Foote et al., "Testing the 400MHz IBM Generation-4 CMOS Chip", Proc. International Test Conference, 1997, pp. 106-114
14. B. Koenemann et al., "Hybrid Random Pattern Self-Testing of Integrated Circuits", US Patent 5612963, 1997
15. M. Kusko et al., "99% AC Test Coverage Using Only LBIST on the 1GHz IBM S/390 zSeries 900 Microprocessor", Proc. International Test Conference 2001, pp.587-592
16. M. Gruetzner and C.W. Starke, "Experience with Biased Random Pattern Generation to Meet the Demands of a High Quality BIST", Proc. European Test Conference 1993, pp. 408-417
17. F. Muradali et al., "A New Procedure for Weighted Random Built-In Self-Test", ITC '90, pp. 660-669, 1990
18. F. Brglez et al., "Hardware-Based Weighted Random Pattern Generation for Boundary Scan", ITC '89, pp. 264-274, 1989
19. B. Koenemann, "Care Bit Density and Test Cube Clusters: Multi-Level Compression Opportunities", submitted to ICCD '03, 2003