

Efficient Test Data Compression Techniques using Viterbi Architecture

P. N. V. A. Swetha¹, P. M. Francis², B. Prasad Kumar³

¹M. Tech, GITAS College, Bibbili, A. P, India

²Department of ECE, Head of Department, Assistant Professor, GITAS, College, Bobbili, A. P, India

³Department of ECE, Assistant Professor, GITAS, College, Bobbili, A. P, India

Abstract: *This paper presents the long data compression and the Cmos transistor count increased new fault technologies these data compression the large data compression and storing the data are described, for provides high encoding efficiency and scalability with respect to the number of test channels support test channel the support algorithm is Viterbi-based test compression. Instead of using the linear equation, the viterbi algorithm proposed scheme finds a set of compressed test vectors very rapidly. According to cost function the branch metric of the Viterbi algorithm, an optimal compressed vector is selected among the possible solution set. By this algorithm we are deriving the condition such as low-power compression and improving capability to repeat test patterns. The proposed on chip de compressor follows the structure of Viterbi encoders which require only one input channel. Experimental results on test volume propose the scheme also yields efficient power-dissipation/volume tradeoff*

Keywords: Logic test, low-power test, on-chip decompressor, scalability, test data compression.

1. Introduction

Test complexity of integrated circuits (ICs) has increased exponentially in scaled CMOS technologies due to increased transistor count and introduction of new fault models relevant for such technologies. Automatic test pattern generators (ATPGs) are widely used to generate the large volume of test data, which is finally stored and applied through an external tester. It should be noted that external testers are “expensive” and hence, low-cost test methodologies for reducing test time and tester memory usage are becoming increasingly important. As a practical solution, researchers have proposed numerous test compression techniques and corresponding architectures that provide comparable fault coverage to that of conventional ATPG and reduce the amount of test data and test time [3]–[9]. Test compression architectures can be broadly classified into three types: code-based, linear-decompression-based, and broadcast-scan-based schemes [1]. Among the above, the linear-decompression-based scheme has gained popularity for simplicity and high encoding efficiency [1]. Linear decompression-based scheme includes a linear decompressor before the scan chains that decompresses the test vectors. Even though different techniques have been suggested for this scheme, successful linear decompressors mainly use linear feedback shift register (LFSR) reseeding [3]–[7] or combinational linear expansion implemented by XOR gates [8]. The basic idea of a linear decompressor is that the test cubes generated by ATPG have a small portion of the bits specified and possible compressed test vectors can be obtained by solving linear equations [3]–[8]. Output response compactor can be designed by using a lossy algorithm, such as multiple input signature registers [1], [9]. Conventional linear decompression architectures find a set of compressed test vectors in a random or heuristic way from the solution space [7]. Hence, they lack the knowledge about the effects of compressed test vectors toward other test

requirements. For instance, such compression may result in high power consumption during the test. Conventional compression architectures may not provide a systematic methodology to trade-off compression with other test requirements. Another issue is designing a scalable (in terms of the number of test channels, where test channel is an external scan input/output port) decompressor. Low pin count test becomes increasingly important due to the limited external interface on a chip and the hierarchical system-on-chip design methodology where each module requires a dedicated test interface [10]. Conventional compression techniques need increasing number of test channels if compression ratio increases [8], [10]. Even though the concept of serializer / deserializer can reduce the number external pins [10], the internal scan configuration is still restricted by the required number of test channels. The key innovation of this paper is the concept of a new linear compression based on the Viterbi algorithm [2], which computes the optimal state sequence in a hidden Markov model, given a sequence of observed outputs. The proposed test data compression architecture is shown in Fig. 1. At every clock cycle, the on-chip Viterbi decompressors accept n bits from the external tester and apply m bits into internal scan chains ($m \gg n$). Assuming scan length to be L , $(L \times n)$ bits are stored in an external tester instead of $(L \times m)$ bits. Thus, we not only have a smaller amount of test data but also a shorter test time. The proposed test compression scheme is highly scalable because each decompressor needs only one test channel and the encoding efficiency is independent of the number of test channels. We can assign any number of test channels for parallel test operation. The proposed compression architecture detects a set of compressed test vectors using the Viterbi algorithm instead of solving linear equations. The branch metric of the Viterbi algorithm includes cost function based on test requirements—e.g., such cost requirements can be based on power dissipation constraints during the test.

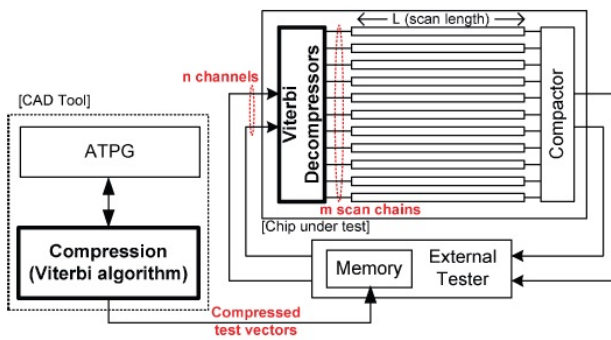


Figure 1: proposed data compression architecture

During the generation of a compressed test vector, the Viterbi algorithm traces the cost function of each possible compressed vector and decides the most desirable compressed vector. The proposed architecture, thus, presents a platform to combine test compression with other test considerations. The rest of this paper is organized as follows.

2. Viterbi Decompressor

The goal of any decompressors in test data compression is to produce large amount of outputs using a small volume of inputs while targeting as many bits to be specified as possible. Because of the large portion of unspecified bits in test cubes, decompressors partly perform as random number generators to fill up unspecified bits in test cubes. It would be interesting to refer to random coding introduced by C. Shannon in 1948 [19]. Shannon claimed that it is possible to transmit information with arbitrarily small error given a noisy channel at a rate less than a certain limit (channel capacity) using random coding (source information is converted into random code words) [19]. Since then all error-correction codes (ECCs) introduced in the literature have been designed in an effort to generate pseudo-random code words with allowable decoding complexity. Considering well-known ECC encoders as test data decompresses would be a reasonable approach to efficient test data compression. Note that conventional ECC encoders in digital communication systems limit the number of outputs to minimize redundant information (i.e., high code rate is preferred). Increasing the number of outputs

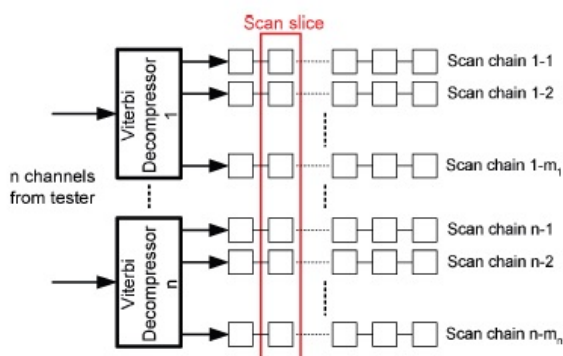


Figure 2: Viterbi decomposition and scan slices

in decompressors, however, is desirable as it leads to better compression ratio (i.e., low code rate is required). In this section, we show that a Viterbi encoder can work as a good decompressor since it has a simple structure with relative ease of increasing the number of outputs. Fig. 2 shows the structure of the proposed on-chip decompressors. Assuming n channels from an external tester, there are n Viterbi decompressors (VDs). At every clock cycle, each decompressor accepts one bit as an input and generates multiple outputs for a scan slice (shown in Fig. 2), which is the set of inputs applied to the scan chain at each scan cycle. The rationale behind this approach is that a test vector generated by ATPG contains a large number of unspecified bits which can be assigned at random. If all unspecified positions in a test vector are left as don't cares, this vector is called a test cube. VDs generate an all-specified test vector by assigning certain values to unspecified positions of a test cube. As a result, compression ratio is determined by the portion of unspecified bits and the capability of decompressors to generate various output combinations for different input sequences. An example VD having four outputs is described in Fig. 3. Each output is the result of two 2-input XOR gates while an input of XOR gate comes from either the channel of a tester or the output of a flip-flop (FF). Each input of an XOR gate is called XOR tap. In this example, all four outputs have three XOR taps and are represented as polynomials or vectors using XOR taps. For example, Out2 has a polynomial of $x_4 + x_3 + 1$

3. Compressed Vector Generation

The Viterbi algorithm is optimal for estimating the state sequence of a finite state process [2]. In general, to find the maximum-likelihood sequence, the Viterbi algorithm constructs a trellis diagram (step 1) and calculates the branch metrics and the path metrics (step 2). Finally, it finds the optimum sequence by a trace back procedure (step 3) [2]. We present examples to illustrate how each step can be utilized for generating compressed test vectors. Given test cubes to be compressed, the Viterbi algorithm traces all possible input sequences. Step 1 (Constructing a Trellis Diagram): in the Viterbi algorithm, a state is represented as a number where the rightmost FF has the most significant bit and the leftmost FF has the least significant bit. Therefore, there are 2^k states, where k implies the number of FFs. In Fig. 3, k is 5 and there are a total of 32 states. At each clock cycle, based on the applied input value, each state generates the corresponding outputs and the next state. This process can be expressed in terms of a trellis diagram, which is a time-indexed version of a state diagram. Fig. 3 depicts a part of the trellis diagram of Fig. 3, where t is the time index. Each transition edge is labeled with corresponding outputs. A trellis diagram can be obtained in any structure that loads inputs at every clock cycle and consists of XOR gates and FFs only.

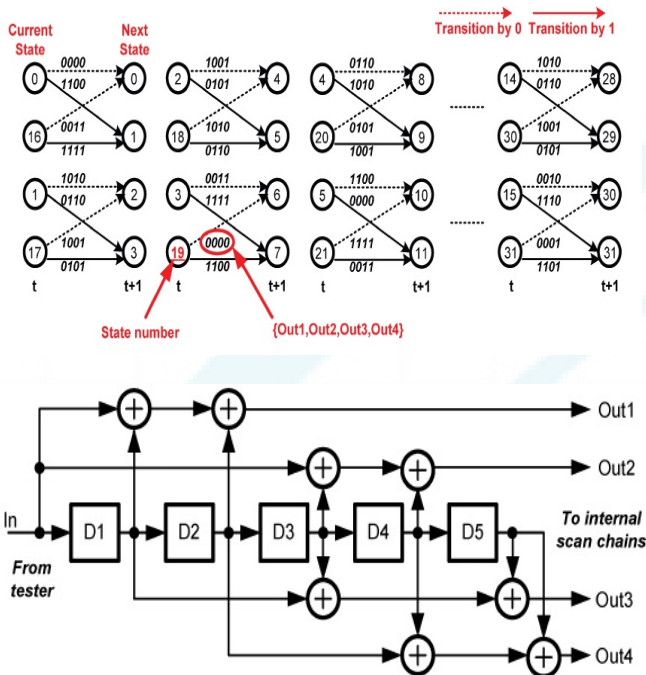


Figure 3: Example of the four –output vertible decompress

4. Efficient Compression Considering Other Test Requirements

In this section, we present two applications of the proposed compression architecture. The branch metric can be defined as a cost function for each feasible transition. By selecting an input sequence of the minimum path metric (at the final time index), we can compress a test cube to meet some of the important test requirements—improving repeat capability and low power consumption during test. Note that such constraints/requirements cannot be easily taken care of by conventional linear compression algorithm which selects a compressed vector at random. It is worth noting that we can assign any test requirements to the branch metric as long as the optimization procedure follows a hidden Markov model. Combining different test requirements is also possible.

A. Improving Repeat Capability Most external testers serve a function of repeating a given test pattern in order to reduce the test data volume [9]. If there is a set of continuously repeated test patterns, then external testers store only one test pattern and keep a binary count of the number of repeated patterns, instead of all patterns. For such a test requirement, we define the branch metric as $\lambda_{i,j}^t = (n \text{ of transitions in the input sequence})$. Using this metric, the Viterbi algorithm selects an input sequence which has minimum transitions among all possible sequences. Even though this branch metric considers only one input channel, minimizing transitions in each channel can increase the chance to increase repeated test patterns for all channels.

B. Low Power Test If unspecified bits in a test vector are loaded with random values; the scan shift operations may cause a prohibitively high switching activity [14]. Note, linear test compression algorithms find a compressed test vector by only considering specified bits. This may result in high power consumption during scan-in shifting. Recently, some test compression architectures that reduce the

transition count have been proposed using additional controllers [13]–[15]. We show that VD can reduce the scan-in power with a simple modification and a proper definition of the branch metric.

The proposed VD structure of Fig. 3 has an important characteristic—it triggers high transition rates in the decompressor outputs for almost all input sequences. As a matter of fact, this is a common feature for test compression since decompressor generates various output patterns for different input sequences. To overcome this problem, we need a method so as not to change the current state at the next time step. The low-power VD (the VD structure that can generate “lowpower” test) has the following definition for the branch metric:

$$\lambda_{i,j}^t = \sum_{p=1}^{N_{out}} (S_{p,t}^i \oplus S_{p,t+1}^j) \times (L + I_n - t - 1)$$

where $S_{p,t}$ is the decompressor output associated with the r^{th} state in the p^{th} scan chain at time index t . This branch metric is able to evaluate shift-in power among all possible input. Other test power constraints can be included by assigning proper cost function to the branch metric if such constraints the hidden Markov model.

5. Experimental Results

The proposed test compression technique was implemented in vhdl programming. The detected faults are removed from the fault list. We detected all detectable faults (100% stuck-at fault coverage). ISCAS89 circuits are small circuits compared to industrial designs. As a result, all ISCAS89 circuits for our experiments have high portion of bits that are specified. In industrial designs which have less than 5% of specified bits [4], the proposed architecture can compress test vectors with much higher ratio.

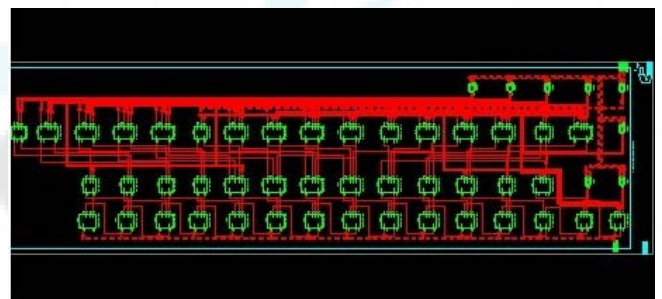


Figure 4: Four- output experimental vertible decompressor diagram

Name	Value	1,999,995 ps	1,999,996 ps	1,999,997 ps	1,999,998 ps	1,999,999 ps	2,000,000 ps
count35	0						
count36	0						
count37	0						
count38	0						
count39	0						
count40	0						
count41	1						
count42	0						
count43	0						
count44	0						
count1	1						
count2	0						
count3	0						

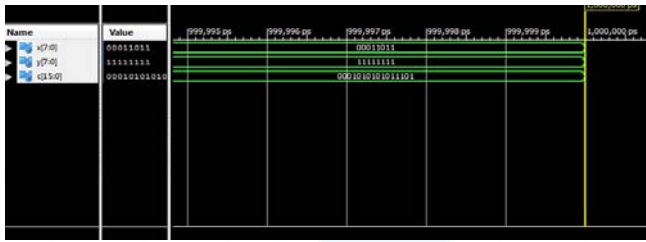


Figure 5: Explains the experimental results

6. Conclusion

In this paper, finally we derived linear compression scheme which is based on the Viterbi algorithm. The proposed technique can include and optimize different test constraints, and selects a set of compressed vectors that has the best cost function among all possible sets. It is creating more rebels and new interesting concepts to digital communication theory ideas to overcome these limitations. Efficient methods for implementing the Viterbi algorithm have been suggested. Some scan chains may have very high number of specified bits and VD may have high probability of not being able to compress these scan chains. The proposed architecture does not require special control signals or modification to the circuit under test. As a result, simple test interface of the proposed scheme significantly reduces the test cost.

References

- [1] N. A. Touba, "Survey of test vector compression techniques," *IEEE Des. Test Comput.*, vol. 23, no. 4, pp. 294–303, Apr. 2006.
- [2] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.
- [3] B. Koenemann, "LFSR-coded test patterns for scan designs," in *Proc. Eur. Test Conf.*, 1991, pp. 237–242.
- [4] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, "Embedded deterministic test," *IEEE Trans. Comput.-Aided Des. Integr. Circuit Syst.*, vol. 23, no. 5, pp. 776–792, May 2004.
- [5] K. J. Balakrishnan and N. A. Touba, "Improving linear test data compression," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 14, no. 11, pp. 1227–1237, Nov. 2006.
- [6] C. V. Krishna, A. Jas, and N. A. Touba, "Reducing test data volume using LFSR reseeding with seed compression," in *Proc. IEEE Int. Test Conf.*, Oct. 2002, pp. 321–330.
- [7] Z. Wang, H. Fang, and K. Chakrabarty, "Deviation-based LFSR reseeding for test-data compression," *IEEE Trans. Comput.-Aided Des. Integr. Circuit Syst.*, vol. 28, no. 2, pp. 259–271, Feb. 2009.
- [8] Bayraktaroglu and A. Orailoglu, "Test volume and application time reduction through scan chain concealment," in *Proc. Des. Autom. Conf.*, 2001, pp. 151–155.
- [9] C. Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, A. Ferko, B. Keller, D. Scott, B. Koenemann, and T. Onodera, "Extending OPMISR beyond 10X scan test efficiency," *IEEE Des. Test Comput.*, vol. 19, no. 5, pp. 65–72, Oct. 2002.
- [10] J. Moreau, T. Droniou, P. Lebourg, and P. Armagnat, "Running scan test on three pins: Yes we can," in *Proc. IEEE Int. Test Conf.*, Nov. 2009, pp. 1–10.
- [11] H. K. Lee and D. S. Ha, "On the generation of test patterns for combinational circuits," Dept. Electr. Eng., Virginia Polytech. Instit. State Univ., Blacksburg, VA, Tech. Rep. 12-93, 1993.
- [12] H. K. Lee and D. S. Ha, "HOPE: An efficient parallel fault simulator," in *Proc. Des. Autom. Conf.*, 1992, pp. 336–340.
- [13] S. Sankaralingam, R. R. Oruganti, and N. A. Touba, "Static compaction techniques to control scan vector power dissipation," in *Proc. IEEE VLSI Test Symp.*, Apr. 2000, pp. 35–40.
- [14] D. Czysz, M. Kassab, X. Lin, G. Mrugalski, J. Rajski, and J. Tyszer, "Low power scan shift and capture in the EDT environment," in *Proc. Int. Test Conf.*, 2008, paper 13.2.
- [15] J. Lee and N. A. Touba, "Low power test data compression based on LFSR reseeding," in *Proc. ICCD*, 2004, pp. 180–185.
- [16] Chandra and K. Chakrabarty, "Frequency-directed run-length (FSR) codes with application to system-on-a-chip test data compression," in *Proc. IEEE VLSI Test Symp.*, May 2001, pp. 42–47.
- [17] G. Mrugalski, J. Rajski, and J. Tyszer, "Ring generators: New devices for embedded test application," *IEEE Trans. Comput.-Aided Des. Integr. Circuit Syst.*, vol. 23, no. 9, pp. 1306–1320, Sep. 2004.