

Cluster Based LFSR Reseeding for Test Data Compression

S. Saravanan, K. Chakrapani and P. Selvakumar
School of Computing, SASTRA University, Thanjavur, India

Abstract: Today's System-on-Chip (SoC) represent high-complexity and it is moving towards the challenge of huge test patterns, more accessing time and larger power consumption. Test data compression is done to improve the test quality. This study presents a test pattern compression by the usage of suitable clustering technique and its corresponding decompression scheme. This scheme includes compression and decompression achieved by LFSR reseeding. Test data compression is widely used in the industry nowadays to reduce the amount of test data stored on the ATE and to decrease testing time. The proposed method requires no special ATPG. The proposed method is validated by the simulation and synthesis output.

Key words: Clustering technique, LFSR reseeding, test data compression

INTRODUCTION

Complexity of an Integrated Circuit has been increased due to continuous scaling of semiconductor technology. However, more faults are likely to occur in the designs as the integration density increases. Due to the complexity of the design, the test data volume also increases rapidly. Since test cost is determined by the test data volume, several commercial tools and various test compression techniques have been developed to reduce both the input test patterns and output responses. Large amount of test data is required for testing complex chips. This large test data exceed the memory capacity of Automatic Test Equipment (ATE). Various techniques have been developed to compress test patterns.

A lot of research is done on lossless compression for test data to reduce the test data volume as it is a major concern. To both test stimuli and test response some of these schemes are applicable, while others consider compression of test stimuli or test response only. In this study, we deal with the amount of test data volume stored in the ATE and how much compression can be achieved.

For complex circuits, longer test time and large tester-memory requirements are the significant causes of increase in test data volume. For a scan-based test, the volume of test data is proportional to the number of scan cells and the number of scan patterns, while test application time can be easily obtained as a ratio of the volume of test data to the number of scan chains divided by the scan-shift frequency. The two major factors that determine the overall test cost are test application time and test data volume. High quality of testing is not guaranteed for chips fabricated with nanometer process. By compressing multiple test cubes into a single seed, compressions achieved by Linear Feedback Shift Register

(LFSR) reseeding can be enhanced. Weight sets are formed by combining multiple test cubes and then compressing the weight set by LFSR reseeding can enhance the compression. This study is an extension of (Saravanan *et al.*, 2011), in which only clustering technique was discussed. But, in this study many techniques such as clustering the test patterns from the ATE, generating a compressed value for each cluster group and regeneration of test patterns by decompression scheme using LFSR reseeding are discussed. Explanation on state diagram, simulation result and targeting FPGA are also provided in this study. The main objective of this study is cluster based LFSR reseeding for test data compression.

EXISTING SYSTEM

Numerous compression techniques have been developed based on LFSR reseeding. LFSR reseeding is also called as linear decompression (Krishna and Touba, 2002; Rajski *et al.*, 2004). By 2013, 1000X compression will be required as predicted by International Technology Roadmap for Semiconductor (IRTS, 2005). Only by LFSR reseeding it is very difficult to achieve 1000X compression. Fault coverage in random pattern based Built-In Self-Test (BIST) can be improved by weighted random pattern testing technique (Bershteyn, 1993). Hence both these techniques can be used to achieve higher compression. Two algorithms are presented for calculation of input weights used during weighted random testing. Those algorithms are based on circuit structure analysis and the pre-generated test patterns to compute weights. The following are some of the advantages of random testing. First, it allows significant compaction of test data which is very much important for BIST.

Secondly, it provides the cost effective means of increasing the coverage of faults not explicitly targeted during deterministic test generation.

In three ways this is a low-cost test strategy:

- Elimination of expensive test pattern computation by the use of random patterns
- The expensive automatic test equipment is replaced by a microcomputer and an ASIC
- Minimization of design for testability overheads unlike other weighted random pattern testing's

The technique of 3-weight weighted random BIST (Hybrid BIST) requires no on-chip memory to store weight sets. Various weights such as 0, 0.25, 0.5 and 0.75, 1 can be assigned to outputs of Test Pattern Generators (TPGs) in conventional weighted random pattern BIST. Only three weights 0, 0.5, 1 are assigned in 3-weight weighted random BIST (Wang, 2001). It can be implemented with low hardware overhead because of its simplicity. Compression achieved by the technique of combining LFSR reseeding and 3-weight weighted random BIST requires two LFSRs, each of which should be loaded with a separate seed for each weight set. Because of this additional compression achieved by this technique is limited. The decompressor proposed here needs only one seed for each weight set to achieve even higher compression by combining LFSR reseeding and 3-weight weighted random BIST. The proposed compression scheme does not require any special Automatic Test Pattern Generator (ATPG) and hence can be used to compress test patterns generated by any ATPG tool.

In low overhead BIST the LFSR is normally used as a test pattern generator. To minimize the switching activity during test application, a test pattern generator called Dual-Speed LFSR (DSLFSR) is used. Two linear feedback shift registers are combined to form DSLFSR namely a normal speed LFSR and a slow LFSR. The normal LFSR is driven by a normal clock. A slow clock, whose speed is $1/d^{\text{th}}$ that of the normal clock is used to drive the slow LFSR. The frequency of transitions at the circuit inputs driven by the slow LFSR is reduced by the usage of DS-LFSR. Hence this leads to reduction in switching during test application (Wang and Gupta, 2002).

In LFSR reseeding, the LFSR is loaded with an initial state called seed. The loaded seed is expanded into a scan pattern when the LFSR is run in an autonomous mode to fill a set of scan chains with a test vector. Size of the scan test pattern is equal to the number of scan flip-flops and the primary inputs of the circuit. The ATE stores only a seed for each scan pattern instead of a complete scan test pattern. Since the size of the seed is often a few orders of magnitude smaller than that of a scan test pattern, the test data volume to be stored in the ATE can be greatly reduced. For implementing LFSR reseeding with seed

compression various architectures have been described (Krishna *et al.*, 2001). The difference in the number of specified bits in each test cube limits the encoding efficiency of the LFSR reseeding. Other than the compression of the test cubes it is a freedom to compress the LFSR seeds to improve the encoding efficiency of LFSR reseeding. It is difficult to change the specified bits in the test cubes and the order of the scan cells in the scan chain. A lot of flexibility is there to organize and align the bits in the seeds so that they can be efficiently encoded with compression codes as there is a large solution space for the linear equations. This scheme of seed compression allows the test data volume for the device under test to be greatly compressed (Hong-Sik and Sungho, 2006). The main advantage of LFSR reseeding techniques is that the scan test patterns have very few specified bits. The bits that are assigned binary values during the test pattern generation are called specified (care) bits. The bits that are not specified are called don't cares.

PROPOSED METHOD

Compression: A test pattern which is having unspecified bits is known as test cube. Compared with the specified bits the amounts of unspecified bits are large in test patterns. Normal form of the test pattern is taken as input for compression. The proposed compression method is to identify the number of conflict bit ('U') in the test data. Test pattern clustering is observed by its threshold value of specified bit with unspecified bit. By dividing the whole test pattern into two groups clustering is done (Saravanan, 2011). One group contains the test patterns with more combination of '0' and '1'. Another group contains the test patterns with more combination of unspecified bits 'X'. Then these two groups are clustered. When there were more specified bits in the test patterns the number of conflicting input bits will be more. If the conflicting bits are less, then it means that more number of unspecified is used for compression and high compression is possible. Thus for reducing the conflict bits proper pattern clustering has to be done. The efficiency of achieving compression based on the multiscan test pattern clustering is high.

The test cube set is nothing but a group of test cubes. These test cubes contains input bits as binary values and don't care values. Consider some test cubes which are of six bit length. Compression takes place and it produces a compressed seed value while combining these test cubes. A circuit with n inputs of a generator is obtained from a set of test cubes and is shown by an n -bit tuple $G^k = (G_1^k, G_2^k, \dots, G_n^k)$, where $G_i^k \in \{0, 1, X, U\}$. The value of p_i is assigned 1 (0) in the corresponding compressed value G^k , if the input p_i is always assigned 'X' or 1 (0) in every test cube in the test cube set and assigned 1 (0) in at least one test cube. Next p_i is assigned 'X' in the corresponding bit if the input p_i is never assigned a binary value '1' or '0' in any test cube of the test cube set. At last, test cube

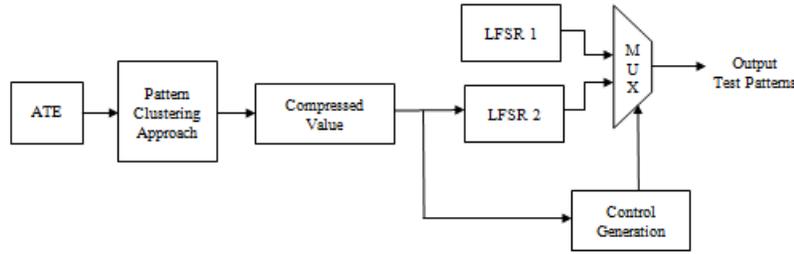


Fig. 1: Block diagram of the system

d^a is said to conflict with another test cube d^b at input test cubes then it is assigned ‘U’ in the compressed generator (G^k) if the input p_i is assigned a 1 (0) in a test cube d^a and assigned a 0 (1) in another test cube d^b in the test cube set. The values that are named ‘U’ in the compressed value G^k are known as conflicting inputs.

Example: In Table 1, the test pattern has ‘X’ or ‘1’ or ‘0’ as its input bits. $D^k = \{d1, d2, d3, d4, d5, d6\}$ is a deterministic test cube set. These test cubes were merged into a single compressed data denoted as generator G^k . The merging operation gives the output as ‘0’ when the entire corresponding bits of the test patterns were ‘0’ or ‘0’ and ‘X’ only. The output will be ‘1’ when the entire corresponding bits of the test patterns were ‘1’ or ‘1’ and ‘X’ only. When the merge is taken between ‘0’ and ‘1’, it denotes ‘U’ which is a conflict value. In the generator G^k the number of conflict value must be $U_{max} \leq 3$. If the number of ‘U’ goes on increasing then more $2^{U_{max}}$ patterns have to be generated. One of the test pattern generated by the decompressor covers the test cube that is merged into the generator G^k .

As only ‘0’ is obtained at p_5 in G^k , weight 0 is assigned to it. Weight 1 is assigned to p_4 and p_2 , as it has ‘1’ and ‘X’. For ‘X’ value, we can consider ‘0’ or ‘1’. But p_1, p_3 and p_6 are assigned ‘0’ in some test cubes and ‘1’ in some other test cubes. So binary values cannot be assigned to these inputs and weight 0.5 is assigned to these inputs and symbol ‘U’ denotes the 0.5 weight.

The block diagram of the scheme shown in Fig. 1, comprises of ATE that contains the test patterns which are used to test the Circuit Under Test (CUT). The test patterns from the ATE are taken and clustering is done as mentioned above. As a result of clustering a compressed value is produced that contains the conflict values in it. The compressed value of the test pattern clusters are called as generator G^k . This compressed result is given as input to the control unit and the LFSR-2. Another test pattern generator namely LFSR-1 is a free running random pattern generator. Depending upon the output of the control unit the values from the LFSR-1 or the values from the LFSR-2 are taken. The output values (S_i) are passed through a scan chain and the test patterns are produced at the end of the scan chain. These test patterns produced are generated from the compressed value.

Table 1: Test data compression method

D^k	P_6	P_5	P_4	P_3	P_2	P_1
d^1	1	×	×	1	×	0
d^2	×	0	×	0	×	0
d^3	1	×	1	0	×	1
d^4	1	×	×	1	×	1
d^5	×	0	×	0	×	1
d^6	0	0	1	1	×	0
G^k	U	0	1	U	×	U

Numerous test patterns will be produced at the end out of which some of the test patterns are the patterns that are taken for producing the compressed value G^k .

Decompression: The decompressor contains an LFSR-1, LFSR-2 and CNTRL. LFSR-1 is a free running random pattern generator. LFSR-2 is a linear test pattern generator. The decompressor also has a multiplexer and a scan chain. The select input of the multiplexer is managed by the control generator. CNTRL stands for selection test pattern generator which comprises of modulo-7 counter, comparator, FIFO and multiplexer. The output of the CNTRL is set to 1 (0) if the input p_i is ‘U’ (0 or 1). Similarly the output of LFSR-1 is 1 (0) if the input p_i is U (X). The values for the inputs that are assigned binary values in G^k are generated by LFSR-2. The output of LFSR-2 is set to a 1 (0) when the p_i is assigned a 1 (0) in G^k . The output of LFSR-2 is ‘X’ if the input p_i are ‘X’ and ‘U’. Based on this the procedure for various test patterns the compressed values are calculated. LFSR-1 and LFSR-2 are the two inputs of the multiplexer. LFSR-1 produces 000, 001, 010, 011, 100, 101, 110 and

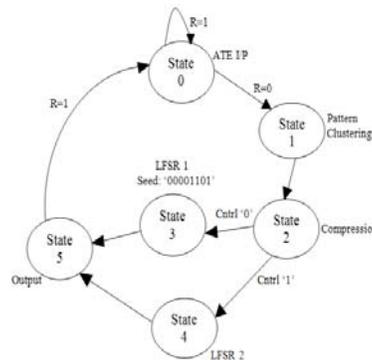


Fig. 2: Proposed state diagram

111 for p_1 , p_3 and p_6 if eight test patterns are generated from G^k . These test patterns detect all faults that are detected by $d_1, d_2, d_3, d_4, d_5, d_6$. The output of the multiplexer S_i is provided to the scan chain which in turn provides the test patterns.

The proposed state diagram in Fig. 2, shows the overall procedure of compression. State 0 represents the initial state which comprises of the ATE. Depending upon the value of reset the next state is reached. Pattern clustering approach is followed in State 1. Clustering of the test patterns obtained from the ATE is done. In State 2 compression of the clustered test patterns is done. Depending on the compressed value, a control signal is generated. If the control value is '0' then it moves to the State 3 which is nothing but LFSR 1 Which has a initial seed value as '00001101'. If the control value is '1' then

State 4 is reached. The State 4 is accompanied by LFSR 2. After this the output test patterns are generated at the State 5. Then the loop is repeated by changing the value of reset signal to 1.

EXPERIMENTAL RESULTS

This section describes the experiment performed to achieve compression of the test patterns based on pattern clustering approach. The proposed method is implemented using VHDL language and its simulation is shown in Fig. 3. The test patterns generated target towards 100% fault coverage. According to the clustering and conflict activity each and every test pattern was compressed. Table 2 shows that the proposed method is targeted towards VIRTEX 2v250fg256 FPGA. Various

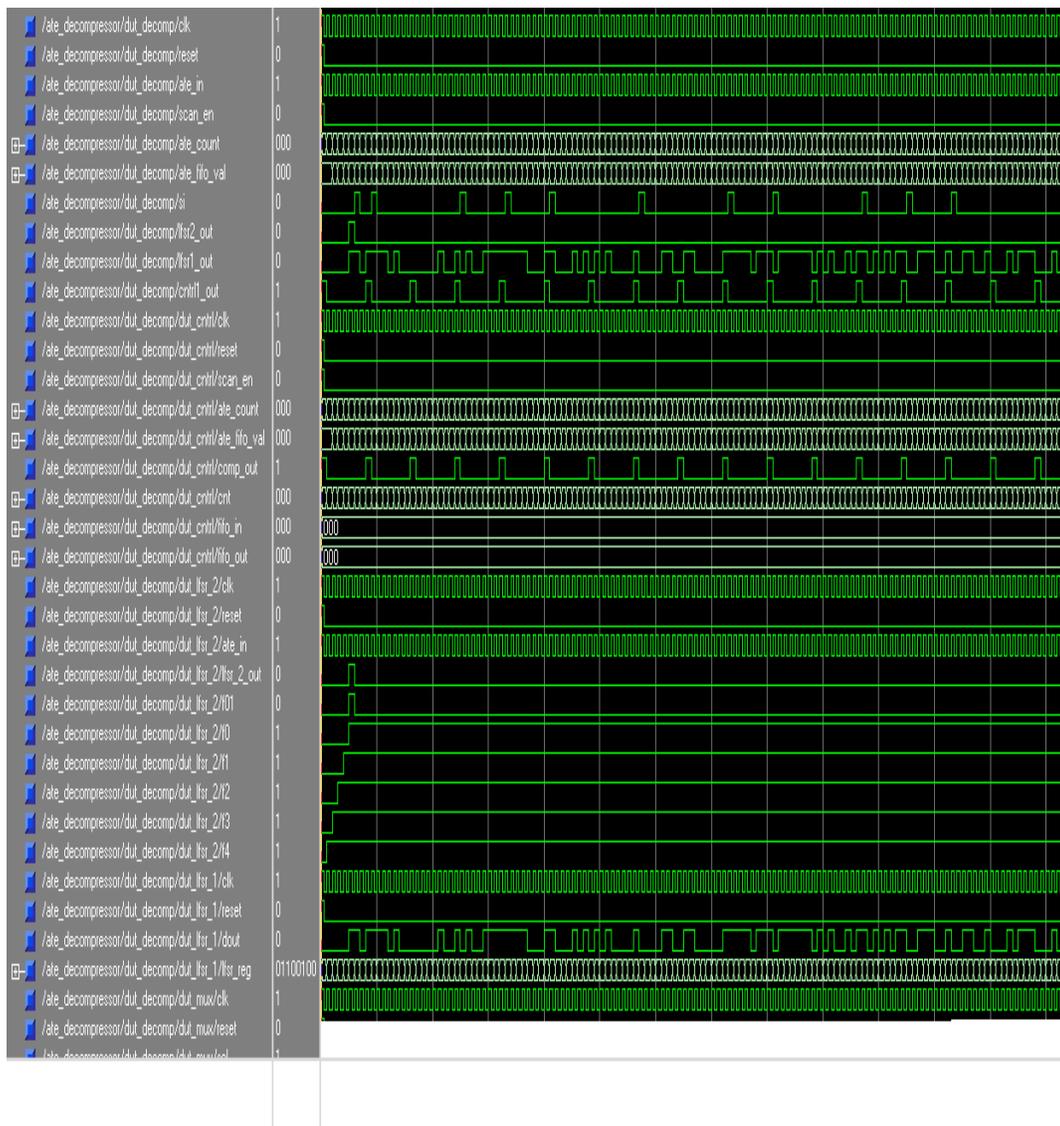


Fig. 3: Simulation result of decompressor

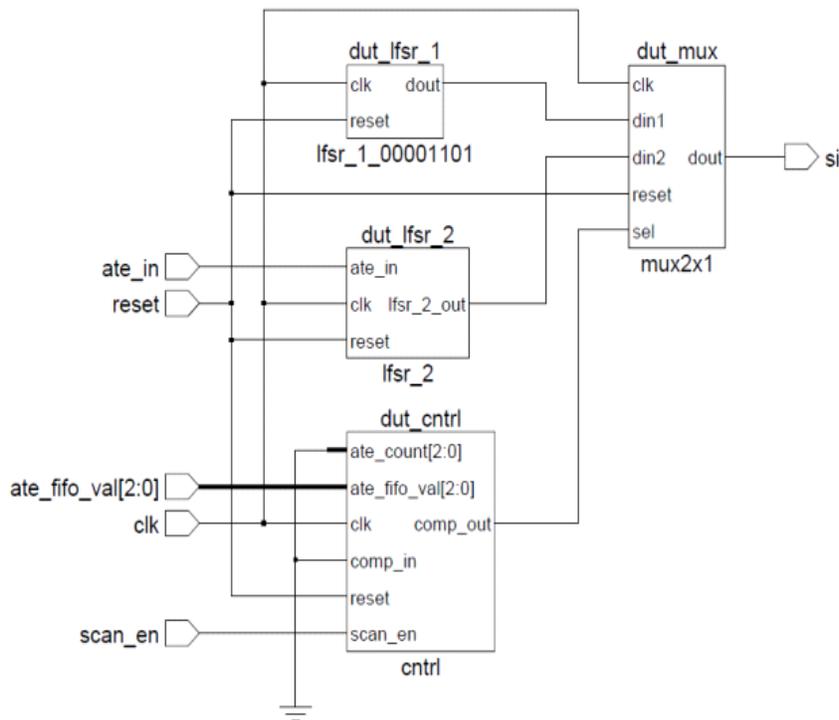


Fig. 4: RTL diagram of decompressor

Table 2: Synthesis report (Device Utilization for 2v250fg256)

Resource	Used	Avail	Utilization
10s	11	172	6.40%
Function Generator	11	3072	0.36%
CLB Slices	14	1536	0.91%
Dffs or Latches	28	3588	0.78%

resources and their utilization are computed from this report. The Register-Transfer Level (RTL) abstraction is shown Fig. 4, to describe the ultimate actual wiring from high level representations of the circuit to the lower level representations. The RTL level design is a typical practice in modern digital design.

ATE is the main source of input. The main blocks of the design are LFSR-1, LFSR-2, CNTRL generation and multiplexer unit. All the blocks are interconnected for logical connections. As LFSR-1 is a free running random pattern generator it is initiated with a seed value. The multiplexer block decides the Si value depending upon the signal from the control unit and the test patterns are obtained.

CONCLUSION

The present challenge of reducing test data is one of the most important tasks in SOC testing. A new method of clustering for the compression of test patterns and a system for the decompressor by which the original compressed test patterns are recovered is proposed in this study. Significant reduction of test patterns is achieved

and the amount of test data to be stored in the ATE for testing complex circuits is also reduced. Proposed method has been validated by the simulation and synthesis results.

REFERENCES

Bershteyn, M., 1993. Calculation of multiple sets of weights for weighted random testing. In: Proceeding International Test Conference, pp: 1031-1040.

International Technology Roadmap for Semiconductors (IRTS), 2005. Test and Test Equipment.

Hong-Sik, K. and S. Kang, 2006. Increasing encoding efficiency of LFSR Reseeding-based test compression. IEEE T. Comput. Aid. D. Integr. Circuits Syst., 25(5): 913-917.

Krishna, C.V. and N.A. Touba, 2002. Reducing test data volume using LFSR reseeding with seed compression. In: Proceeding International Test Conference, pp: 321-330.

Krishna, C.V., A. Jas and N.A.Touba, 2001. Test vector encoding using partial LFSR reseeding. Proceeding ITC, pp: 885-893.

Rajski, J., J. Tyszer, M. Kassab and N. Mukherjee, 2004. Embedded deterministic test. IEEE T. Comput. Aid. D. Integr. Circuit Syst., 23(5): 776-792.

Saravanan, S., P. Selvakumar, A. Balasubramanian and R. Silambamuthan, 2011. Achieving higher test data compression using pattern clustering technique. In: Proceeding IEEE ICCIC., pp: 484-487.

Wang, S., 2001. Low hardware overhead scan based 3-weight weighted random BIST. In: Proceeding International Test Conference, pp: 868-877.

Wang, S. and S.K. Gupta, 2002. DS-LFSR: A BIST TPG for low switching activity. IEEE T. Comput. Aid. D. Integr. Cir. Syst., 21(7): 842-851.