湖南大学学报（自然科学版）
Journal of Hunan University (Natural Sciences)

第 50 卷 第 1 期
2023 年 1 月

Available online at http://jonuns.com/index.php/journal/index

Vol. 50 No. 1
January 2023

Open Access Article https://doi.org/10.55463/issn.1674-2974.50.1.23

Validation of Proposed Test Set Reduction Hybrid Compaction Schemes for FPGA-Based Designs

Abdul Rafay Khatri*

Department of Electronic Engineering, Quaid-e-Awam University of Engineering, Science and Technology, Nawabshah, Sindh, Pakistan

* Corresponding author: arkhatri@quest.edu.pk

Received: December 27, 2022 / Revised: January 21, 2023 / Accepted: January 30, 2023 / Published: February 28, 2023

Abstract: Recently, the density and intricacy of Very Large-Scale Integration (VLSI) circuits are grown, and the number of test vectors has increased drastically. Owing to this, the cost of testing is a significant component for consideration. The two main factors that measure the cost of the test are the size of the test vector sets and test data volume. Compression and compaction are the two approaches used to reduce the aforementioned factors. Compaction is a primary process in the test generation procedure, and there are two classes for it, i.e., dynamic and static compaction approaches. In this paper, the author performs testing and obtains the compact test vectors for various FPGA-based designs. The RASP-FIT fault injection testing tool provides the ATPG testing framework. In this paper, the testing is performed on various FPGA benchmark designs written in Verilog HDL, and the results are validated. The ATPG method developed under the RASP-FIT tool contains new hybrid compaction techniques that calculate compaction and fault coverage for the designs. This work is also compared with the previous work and found that the proposed compaction scheme is better at compacting the test vectors.

Keywords: compression, dynamic compaction, static compaction, test generation, test vector size.

基于 FPGA 设计的拟议测试集减少混合压缩方案的验证

摘要：最近，超大规模集成（超大规模集成电路）电路的密度和复杂性都在增加，测试载体的数量也急剧增加。由于这个原因，测试成本是需要考虑的一个重要组成部分。衡量测试成本的两个主要因素是测试向量集的大小和测试数据量。压缩和压实是用于减少上述因素的两种方法。压实是测试生成程序中的一个主要过程，它有两类，即动态和静态压实方法。在本文中，作者对各种基于 FPGA 的设计进行了测试并获得了紧凑的测试向量。锉刀配合故障注入测试工具提供了 ATPG 测试框架。在本文中，对用高密度脂蛋白语言编写的各种 FPGA 基准设计进行了测试，并对结果进行了验证。在锉刀配合工具下开发的 ATPG 方法包含新的混合压实技术，计算设计的压实度和故障覆盖率。这项工作还将结果与以前的工作进行了比较，发现所提出的压实方案在压实测试向量方面效果更好。

关键词：压缩，动态压实，静态压实，测试生成，测试向量大小。

1. Introduction

The complexity of Very Large-Scale Integration (VLSI) designs is continuously growing with the significant development in science and technology and shrinking transistor feature size. The probability of faults occurring also increases with the increasing density and complexity of the models. Digital design testing is, therefore, necessary to ensure the fault-free operation of devices [1]. The testing cost for such systems is also increasingly growing because of the complexity of VLSI circuits [2]. Several methods for analyzing digital systems have been proposed and are widely known as methods of producing test patterns. The primary objective of testing approaches is to confirm the device free from faults or defects, to comply with the claimed requirements, and to improve the yields of the device [3]. The test procedure generates test vectors for automated test equipment, and they should be compacted for memory efficiency. Two techniques, compression and compaction, are used to decrease the test set size [4]. Compression of the set of test vectors is a hardware-based technique.

It compresses and decompresses the test stimuli and test responses, respectively, using the on-chip hardware [4]. To save test data, various test compaction techniques are used to decrease the test pattern count (while the value of Fault Coverage (FC) is maintained). These techniques are named as dynamic and static compaction schemes [5].

During the last few decades, Hardware Description Languages (HDLs) have been actively participating in enhancing different methods of designing and testing digital systems. Because of HDLs, the design and test engineers work closely as it improves the coordination between the methods and techniques used by them. The design engineers may review and test the designs during the initial phase of the design flow in HDL. In addition, the designs are tested without translating into any compatible format [6]. Digital system testing is done at the beginning so that the time to market, cost, and effort is reduced in the designing process. To perform Fault Injection (FI) testing, the test engineer requires two modules,

- A golden reference module (aka fault-free circuit)
- A faulty module (in which faults are introduced deliberately aka defective design/circuit)

Test engineers need at least the following steps to test the System under Test (SUT). These steps are given in the sequel:

- To develop a mechanism to generate a faulty module from the original golden module.
- To inject faults in all possible locations of the design.
- A mechanism that both randomly or sequentially selects and activates the injected faults in the target design.
- Finally, perform fault injection testing and calculate many parameters such as fault coverage (FC) and compaction ratio (C).

To meet the following conditions, a simple test pattern generation is needed for the FPGA designs. Hence, a new automatic method is developed and presented in our previous work [3], [6]–[8]. This ATPG test method proposes hybrid compaction methods to minimize the test vector count and to obtain compact test vectors for FPGA-based systems. The test method should contain the proposed dynamic compaction scheme, whereas, the proposed static compaction is developed under the RASP-FIT tool [6]. In this paper, the main objective is to validate the claims made for the proposed test and compaction techniques under the RASP-FIT tool developed previously in the continuation of this research [8].

The whole paper is further structured as follows: Section 1 discusses state-of-the-art techniques for dynamic and static compaction approaches developed in the last few decades. The proposed dynamic and static techniques are described in Section 2 in detail. The ATPG method with the proposed compaction schemes is applied to various benchmark designs, and the results are evaluated. Section 3 discusses the results. After that, Section 4 compares the results presented in this paper with the state-of-the-art research available in the literature. Section 5 concludes the paper and recommends the next steps for advanced research.

2. Related Work

Compact test sets are highly suitable for modern VLSI designs to decrease test implementation time and memory usage. Recently, high-quality test sets are receiving great attention for increasing test quality and performance. The International Technology Roadmap for Semiconductors (ITRS) predicts the requirement of compression and described a thousand times the compression factor needed till 2020. Test compaction techniques are therefore necessary to obtain a highly qualified test vector [9]. Test generation methods are primarily used to acquire fewer test vectors for the maximum coverage of faults, and it is called the test compaction procedure. These procedures or algorithms minimize the test set size. Therefore, reducing the time of application of the test and the volume of test data [8], [10].

The main objective is to reduce the memory requirement for Automated Test Equipment (ATE). Due to this, we achieve faster operation of ATE. For

this function, two techniques are usually used: compression and compaction [3], [5]. In this research work, the compaction method is carried out, which is further categorized into two and explained briefly in the sequel:

- Dynamic Compaction
- Static Compaction

2.1. Dynamic Compaction Schemes

The dynamic compaction techniques are part of an ATPG process that means it is a technique or method that generates the compact test vector during the ATPG process. Various techniques are available in the literature that try achieving the compact test vectors such as separate fault sets and their compatibility [11]–[13], reverse -order fault simulation [14], double detection [15], Reverse Order Test COmpaction (ROTCO), forward-looking reverse order fault simulation [16], and rotating back-trace [8], [17].

Xiang et al. [18] presented the work on a new dynamic test compaction approach. They proposed a new calculation that estimates the impact inputs of a transition fault, a subset of inputs that can be defined to detect the transition fault, which is used by the dynamic test compaction scheme. This attempts to compress experiments into this test with as many defects as possible. Authors in [1], and [4] indicate an incremental dynamic compaction technique. A cube merging technique is used with dynamic compaction in this process.

The Parallel Order Dynamic Test Compaction (PODTC) technique shows that the order of secondary faults is vital for test compaction within a single test generation. This method launches parallel ATPG and selects the best pattern that found the critical faults [9]. A new method for checking compaction has recently been proposed by [10] under a transparent scan.

2.2. Static Compaction Schemes

In the static compaction method, the quantity of test vectors is diminished after they are created. The static compaction strategies applied after the generation of test vectors. Therefore, these techniques are executed as an independent project and are free of ATPG approaches. These techniques are effortlessly consolidated into a test environment [19].

Static compaction [20]-[21] is, therefore, less expensive than that dynamic compaction. For combinational circuit test compaction, several techniques are available. These techniques are used in combination to implement a test compaction program for combinational circuits, namely,

- Redundant Test Vector Elimination Method
- Test Vector Merging
- Test Subset Merging
- A Multi-cut Test Set Merging
- Test Vector Modification
- Reverse Order Fault Simulation
- Proposed Compaction Schemes

Several methods for analyzing the FPGA-based digital systems have been proposed and are widely known as methods of producing test patterns. The object of these methods of testing is to ensure that the device is free of defects, to comply with the claimed requirements and to boost yields. The key objectives of the TPG are [22]:

- To find compact test vectors;
- To obtain a memory efficient solution;
- To achieve faster operation by fastening the test procedure;
- To be a cost-effective method.

The hybrid compaction schemes were developed under the RASP-FIT fault injection tool. The author developed this tool at the University of Kassel, Germany. More details about the tool are given in [3], [6], [7], [23].

3. Methodology

3.1. Proposed Dynamic Compaction

As described earlier, the dynamic compaction technique is part of the ATPG method. The implementation of the proposed dynamic compaction scheme and its Verilog HDL code is generated by the RASP-FIT tool. This code is added to the top_file automatically [3], [6]. The dynamic compaction scheme introduces a concept based on the set-point value. The set-point value is a number that is the mean of observation points obtained during fault injection experiments. For more information on how to find the set-point values, refer to [24].

3.1.1. Memory Unit

After the dynamic compaction is performed, the test vectors are stored in the memory unit defined in the Verilog HDL code by the RASP-FIT tool. The parameters are added to the “top file” that decides the size of the memory. The memory unit consists of columns and rows. The columns show the total number of faults injected in the design using the fault injection experiment, whereas the rows show the test vector count obtained during the dynamic compaction process. The resulting data is stored in the text (*.txt) file and applied to the result analyzer, developed under the RASP-FIT tool, for static compaction.

3.2. Proposed Static Compaction

As described earlier, static compaction is not a part of the ATPG system. It decreases the number of test vectors further after their generation. Inside the RASP-FIT tool, developed by the author [8], a simple static compaction scheme was proposed, which is memory-efficient and fast. It is written in MATLAB, which takes a few input parameters, and data files and works on that data. The proposed scheme measures the coverage and compaction of faults and reduces the number of test

vectors. The tab for static compaction is shown in Figure 1. All input parameters needed for compact test vector calculation and fault coverage are defined briefly as follows [25]:

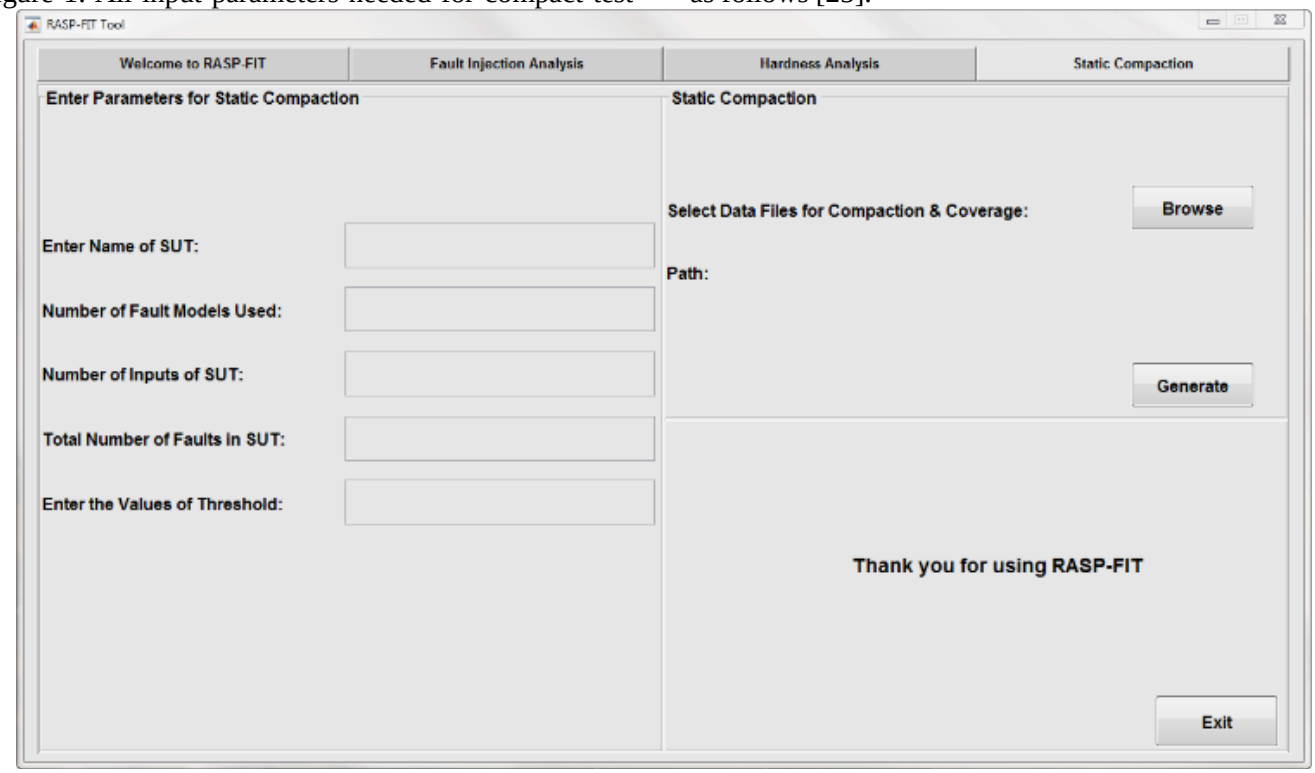

Fig. 1 Static compaction tab under RASP-FIT tool [8]

- The name of the SUT: This parameter takes the name of the file that contains generated data after static compaction is successfully run.
- Number of fault models used: This parameter requires the total number of fault models used (e.g., 3).
- The number of inputs of SUT: This parameter requires information about the number of inputs of the design under investigation.
- The total number of faults in SUT: How many faults are injected in the SUT? This information is critical for the entire procedure.
- The threshold value: The user specifies threshold values to separate the vulnerable positions into hard-to-find and most commonly found.

The data files are loaded for static compaction into the RASP-FIT tool. These files were used during the data generation process to further decrease the number of test vectors. The fault coverage and compaction ratio are also computed by the static compaction. The static compaction scheme in the RASP-FIT tool reads the *.txt file and positions the data in a matrix form, as given in Eq. 1.

$$FM = \begin{bmatrix} P_1 & F_{1,1} \cdots & F_{N,1} \\ P_2 & F2_{1,2} \cdots & F_{N,2} \\ P_i & F_{1,i} \cdots & F_{N,i} \end{bmatrix} \quad (1)$$

where P_1—P_i represent the qualified test pattern obtained after dynamic compaction, and the $F_{N,i}$ is the fault detection value for the N^{th} fault on an i^{th} qualified pattern. The value is `1' for $F_{N,i}$ shows the detection of faults and otherwise, it is not detected. This matrix is used to obtain compact test vectors for the maximum FC in static compaction. The following steps describe the main components of static compaction in detail [3], [7].

3.2.1. Fault Counting

Data are formatted as the pattern and fault detection array, as shown in Eq. 1. Input pattern length is provided as an input to the RASP-FIT tool, which removes the input pattern from the data.

For each fault array, the number of fault detection (the value, which is 1' in the fault matrix) are counted and the sum is indexed as a column vector.

3.2.2. Sorting

In the previous step, we counted the number of faults found in each pattern. In this step, sorting occurs. The sorted numbers are placed in descending order.

3.2.3. Merging

Merging is the process to combine two vectors. Sorted patterns are stored in a memory. After the sorting step, the first vector has the highest fault detection count, and the proposed merger method performs a logical OR operation with the next vector below it. This step continues till all the vectors obtained during the fault injection testing with dynamic compaction are checked.

3.3. Fault Coverage Computation

Each time, the FC is computed and compared with its previous value. After the comparison, it decides whether the test vector should be considered for the

inclusion in the compact test vector list or not. If the value of FC is increased from its previous value, it will be added to the list. In another case, this test vector is removed. Fault coverage is computed mathematically as,

$$FC\ (\%) = \frac{F_D}{F_T}\ X\ 100 \qquad (2)$$

where FC stands for Fault Coverage, FD shows the fault detection count, and FT represents the total injected faults.

3.4. Compaction Ratio Computation

By definition, the compaction ratio is defined as "the minimum number of vectors obtained to reach the maximum FC of a circuit".

$$C\ (\%) = \left(1 - \frac{T_{static\_compact}}{T_q}\right) X\ 100 \qquad (3)$$

where C stands for the compaction, $T_{static\_compact}$ represents a number of compact test vectors, and T_q represents the qualified test vectors (dynamic compaction scheme).

4. Results and Discussion

The proposed test approach with its compaction schemes is applied to various benchmark circuits written at different abstraction levels such as gate level, data flow level, and behavioral level. The results are presented in Table 1, which shows the total number of injected faults by the automatic fault injection tool, i.e., RASP-FIT. Also, the test method gives a fault detection count, and the fault coverage is then evaluated by Eq. (2) [3], [8]. The compaction ratio is also evaluated for all systems under test by Eq. (3) for the benchmark designs and the results are presented in Table 2. The more value of the compaction ratio gives a smaller number of test vectors.

Table 1 Fault detection and fault coverage for FPGA (Verilog HDL) designs

FPGA Design	Total Fault Injected (T_F)	Faults Detected (F_D)	Fault Coverage (%)
ISCAS Circuit-17	12	12	100
ISCAS Circuit-432	336	333, 328, 324	99.1,97.6,96.4
ISCAS Circuit-499	408	406	99.5
ISCAS Circuit-880	729	729	100
ISCAS Circuit-1355	1064	1064, 1046, 1030	100, 98.3, 96.8
ISCAS Circuit-1908	1498	1430, 1395, 1367	95.4, 93.1, 91.2
ISCAS Circuit-2670	2152	2110, 2074, 1941	98.05, 96.3, 90.2
EPFL Adder	2040	2036	99.85
EPFL Shifter (Barrel)	6672	6669	99.96
EPFL Sine ckt	10832	9867,9401,8976	91,86.7,82.8
EPFL ALU Unit	349	346	98.85
EPFL Decoder ckt	608	602,599,230	99,98.5,38
Integer-to-Float	520	337,264,196	64,50.7,37.6
EPFL IIC Controller	2699	2206,2071,1936	81.7,76.7,71.3
EPFL Priority Encoder	1956	1550,1359,1410	79.2,69.4,72.09
EPFL Coding-Cavlc	1386	1380	99.5
Circuit Bitwise	09	09	100
Boolean ckt	10	10	100
Adder (32-bit)	194	194	100
Mux case	24	24	100

Table 2 Hybrid compaction for FPGA designs in Verilog HDL

FPGA Design	Compaction (Dynamic)	Compaction (Static)	Compaction (%) Bit-flip,SA-1,SA-0
ISCAS Circuit-17	32	2, 5, 7	93.8, 84.5, 78.13
ISCAS Circuit-432	100	28, 42, 59	72, 58, 41
ISCAS Circuit-499	100	11, 14, 42	89, 86, 58
ISCAS Circuit-880	100	30, 47, 63	70, 53, 37
ISCAS Circuit-1355	100	28, 38, 39	72, 62, 61
ISCAS Circuit-1908	100	53, 40, 50	47, 60, 50
ISCAS Circuit-2670	100	38, 43, 60	62, 57, 40
EPFL Adder	100	12,20,27	88,80,73
EPFL Shifter (Barrel)	100	23,34,54	77,66,46
EPFL Sine ckt	100	57,86,83	43,14,17
EPFL ALU Unit	100	20,23,34	80,77,66
EPFL Decoder ckt	100	22,38,63	78,62,37
Int-to-Flt	100	6,6,10	94,94,90
EPFL I2C Controller	100	50,52,85	50,48,15
EPFL Priority Encoder	100	44,59,34	54,41,66
Coding-Cavlc	100	71,94,75	29,6,25
Circuit Bitwise	100	2	98
Boolean ckt	100	2,3,3	98,97,98
Adder (32-bit)	100	2	98
Mux case	100	3,5,4	97,95,96

Table 3 Comparison between state-of-the-art works with the proposed tool

System Under Test	Work 1 [18]		Work 2 [7]		Work 3 [4]		Work 4 [28]		Proposed Work	
	TV	FC	TV	FC	TV	FC	TV	FC	TV	FC
ISCAS Circuit-17	30	100	6	-	-	-	7	100	2,5,7	100
ISCAS Circuit-432	360	96.6	57	-	32	98.7	54	99.24	28,42,59	99.1,97.6,96.4
ISCAS Circuit-499	340	97.6	67	-	52	98.6	53	100	22,29,42	99.5
ISCAS Circuit-880	600	97.3	66	-	25	100	60	100	30,47,63	100
ISCAS Circuit-1355	500	96.2	96	-	84	99.5	85	100	28,39,39	100, 98.3, 96.8
ISCAS Circuit-1908	500	91.6	142	-	-	-	114	99.89	53,40,50	95.4, 93.1, 91.2
ISCAS Circuit-2670	600	82	259	-	-	-	107	98.84	38,43,60	98.05, 96.3, 90.2

The results on various FPGA designs are compared with previous test methods available in the literature and compared in Table 3. The results are compared with the state-of-the-artwork such as the simulation-based method presented in [26]. Another work is SAT {SATisfiability}-ATPG with a dynamic test vector compaction approach, which was presented in [27].

Similarly, [28] and [29] are presented for some benchmark designs. The last columns of the table show that the proposed methodology achieved fewer test vectors for the more significant value of fault coverage [8]. No comparison is available for the other benchmark designs in the literature. From the results presented in Table 3, it is clearly shown that the proposed compaction method has achieved better results compared to other work in terms of fault coverage and the number of test vectors.

5. Conclusion

In this paper, the proposed hybrid compaction schemes under the RASP-FIT fault injection tool are validated. This tool is designed to perform the ATPG approach and obtains the compact test vectors for Verilog designs (e.g., FPGA designs). The main findings of this research are to obtain the compact test vectors for various FPGA designs and to validate the proposed compaction schemes. The compaction of test vectors is an utmost important issue for the memory efficiency of automatic test equipment (ATE). The fault injection technique injects faults at every abstraction level of the design process, i.e., automatically modifies the Verilog code during the test pattern generation step. Many FPGA designs are tested, and compact test vectors are obtained. Comparison with other studies showed that this work is better than the state-of-the-art research available in the literature and found that our proposed method performed better compaction. In future research, it is recommended that other fault models must be developed and tested.

References

[1] JHA S. Compaction mechanism to reduce test pattern counts and segmented delay fault testing for path delay faults. University of Iowa, 2013.

[2] AHMADY M, and SAYEDI S M. Fault coverage improvement and test vector generation for combinational circuits using spectral analysis. *Proceedings of the 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2012, 1–5.

[3] KHATRI A R, HAYEK A, and BÖRSCÖK J. ATPG method with a hybrid compaction technique for combinational digital systems. *Proceedings of the SAI Computing Conference (SAI)*, 2016, 924–930.

[4] JHA S, CHANDRASEKAR K, WU W, et al. A Cube-Aware Compaction Method for Scan ATPG. *Proceedings of the 27th International Conference on VLSI Design and 13th International Conference on Embedded Systems*, 2014, 98–103.

[5] EGGERSGLUB S, SCHMITZ K, KRENZ-BAATH R, and DRECHSLER R Optimization-based multiple target test generation for highly compacted test sets. *Proceedings of the 19th IEEE European Test Symposium (ETS)*, 2014, 1–6.

[6] KHATRI A R, HAYEK A, and BÖRSCÖK J. RASP-FIT: A Fast and Automatic Fault Injection Tool for Code-Modification of FPGA Designs. *International Journal of Advanced Computer Science and Applications*, 2018, 9(10), 30–40.

[7] KHATRI A R, HAYEK A, and BÖRSCÖK J. Validation of the Proposed Fault Injection, Test and Hardness Analysis for Combinational Data-Flow Verilog HDL Designs Under the RASP-FIT Tool. *Proceedings of the IEEE 16th International Conference on Dependable, Autonomic and Secure Computing*, 2018, 544–551.

[8] KHATRI A R. *Development, Verification and Analysis of a Fault-Injection Tool for improving Dependability of FPGA Systems*. University of Kassel, 2019.

[9] CHEN Y-W, HO Y-H, CHANG C-M, et al. Parallel order ATPG for test compaction. *Proceedings of the International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, 2018, 1–4.

[10] POMERANZ I. Test Compaction by Test Removal Under Transparent Scan. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2019, 27(2), 496–500.

[11] KAJIHARA S, POMERANZ I, KINOSHITA K, and REDDY S M. Cost-effective generation of minimal test sets for stuck-at faults in combinational logic circuits. *Proceedings of the 30th international on Design automation conference (DAC '93)*, 1993, 102–106.

[12] REDDY L N, POMERANZ I, and REDDY S M. COMPACTEST-II: a method to generate compact two-pattern test sets for combinational logic circuits. *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, 1992, 12(7), 568–574.

[13] TROMP G. Minimal Test Sets for Combinational Circuits. *Proceedings of the International Test Conference*, 1991, p. 204.

[14] SCHULZ M H, TRISCHLER E, and SARFERT T M. SOCRATES: a highly efficient automatic test pattern generation system. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1988, 7(1), 126–137.

[15] KAJIHARA S, POMERANZ I, KINOSHITA K, and REDDY S M. Cost-effective generation of minimal test sets for stuck-at faults in combinational logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1995, 14(12), 1496–1504.
[16] POMERANZ I, REDDY S M, and LIN X. Experimental results of forward-looking reverse order fault simulation on industrial circuits with scan. *Proceedings of the 10th Asian Test Symposium*, 2001, p. 467.
[17] REDDY L N, POMERANZ I, and REDDY S M. ROTCO: a reverse order test compaction technique. *Proceedings of the Euro ASIC '92*, 1992, 52242, 189–194.
[18] XIANG D, SUI W, YIN B, and CHENG K. Compact Test Generation With an Influence Input Measure for Launch-On-Capture Transition Fault Testing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2014, 22(9), 1968–1979.
[19] NAVABI Z. *Digital System Test and Testable Design*. Boston, MA: Springer US, 2011.
[20] EL-MALEH A H, KHURSHEED S S, and SAIT S M. Efficient Static Compaction Techniques for Sequential Circuits Based on Reverse Order Restoration and Test Relaxation. *Proceedings of the 14th Asian Test Symposium (ATS'05)*, 2005, 378–385.
[21] DIMOPOULOS M, and LINARDIS P. Efficient static compaction of test sequence sets through the application of set covering techniques. *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, 2004, 194–199.
[22] SAHARI M S, A'AIN A K, and GROUT I. A study on the effect of test vector randomness on test length and its fault coverage. *Proceedings of the 10th IEEE International Conference on Semiconductor Electronics (ICSE)*, 2012, 503–506.
[23] KHATRI A R, HAYEK A, and BÖRSCÖK J. Fault Injection and Test Approach for Behavioural Verilog Designs using the Proposed RASP-FIT Tool. *International Journal of Advanced Computer Science and Applications*, 2019, 10(4), 57–63.
[24] KHATRI A R, HAYEK A, and BÖRSCÖK J. Validation of selecting SP-values for fault models under proposed RASP-FIT tool. *Proceedings of the First International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT)*, 2017, 1–7.
[25] KHATRI A R. A Technical Guide for the RASP-FIT Tool. *International Journal of Advanced Computer Science and Applications*, 2019, 10(12), http://dx.doi.org/10.14569/IJACSA.2019.0101279
[26] MIRKHANI S, and ABRAHAM J A. Fast evaluation of test vector set using a simulation-based statistical metric. *Proceedings of the IEEE 32nd VLSI Test Symposium (VTS)*, 2014, 1–6.
[27] HABIB K, SAFAR M, DESSOUKY M, and SALEM A. Don't care based dynamic test vector compaction in SAT-ATPG. *Proceedings of the IEEE 57th International Midwest Symposium on Circuits and Systems*, 2014, 213–217.
[28] DUNBAR C, and NEPAL K. Using Platform FPGAs for Fault Emulation and Test-set Generation to Detect Stuck-at Faults. *Journal of Computers*, 2011, 6(11), 2335–2344.
[29] ZHANG Y. *Diagnostic Test Pattern Generation and Fault Simulation for Stuck-at and Transition Faults*. Auburn University, 2012.

参考文:

[1] JHA S. 用于减少路径延迟故障的测试模式计数和分段延迟故障测试的压缩机制。爱荷华大学，2013。
[2] AHMADY M 和 SAYEDI S M. 使用频谱分析改进组合电路的故障覆盖率和测试向量生成。第 25 届 IEEE 加拿大电气与计算机工程会议(中国建筑工程学院)论文集，2012，1–5。
[3] KHATRI A R、HAYEK A 和 BÖRSCÖK J. ATPG 方法和用于组合数字系统的混合压实技术。SAI 计算会议(SAI)论文集，2016，924–930。
[4] JHA S、CHANDRASEKAR K、WU W 等。一种用于扫描 ATPG 的立方体感知压缩方法。第 27 届超大规模集成电路设计国际会议和第 13 届嵌入式系统国际会议论文集，2014，98-103。
[5] EGGERSGLUB S、SCHMITZ K、KRENZ-BAATH R 和 DRECHSLER R 基于优化的多目标测试生成，用于高度压缩的测试集。第 19 届 IEEE 欧洲测试研讨会(碳排放交易体系)论文集，2014，1-6。
[6] KHATRI A R、HAYEK A 和 BÖRSCÖK J. 铿刀配合：一种用于 FPGA 设计代码修改的快速自动故障注入工具。国际高级计算机科学与应用杂志, 2018, 9(10), 30–40.
[7] KHATRI A R、HAYEK A 和 BÖRSCÖK J。在铿刀配合工具下对组合数据流高密度脂蛋白语言设计的拟议故障注入、测试和硬度分析进行验证。IEEE 第 16 届可靠、自主和安全计算国际会议论文集，2018，544–551。
[8] KHATRI A R. 用于提高 FPGA 系统可靠性的故障注入工具的开发、验证和分析。卡塞尔大学，2019。
[9] CHEN Y-W, HO Y-H, CHANG C-M, 等. 用于测试压实的并行顺序 ATPG。超大规模集成电路设计、自动化和测试(超大规模集成电路)国际研讨会论文集，2018，1-4.
[10] POMERANZ I. 通过透明扫描下的测试移除测试压实。IEEE 超大规模集成(超大规模集成电路)系统汇刊, 2019, 27(2), 496–500。
[11] KAJIHARA S、POMERANZ I、KINOSHITA K 和 REDDY S M。为组合逻辑电路中的固定故障生成具有成本效益的最小测试集。第 30 届国际设计自动化会议论文集(解码器'93)，1993，102–106。
[12] REDDY L N、POMERANZ I 和 REDDY S M. 最紧凑-II：一种为组合逻辑电路生成紧凑型双模式测试集的方法。IEEE/美国计算机学会计算机辅助设计国际会议论文集，1992，12(7)，568–574。
[13] TROMP G. 组合电路的最小测试集。国际测试会议记录，1991, 204.
[14] SCHULZ M H、TRISCHLER E 和 SARFERT T M. 苏格拉底：一种高效的自动测试模式生成系统。IEEE 集成电路和系统计算机辅助设计汇刊，1988，7(1)，126–137。
[15] KAJIHARA S、POMERANZ I、KINOSHITA K 和 REDDY S M。为组合逻辑电路中的固定故障生成具有成本效益的最小测试集。IEEE 集成电路和系统计算机辅助设计汇刊，1995，14(12)，1496-1504。
[16] POMERANZ I, REDDY S M, 和 LIN X. 前瞻性工业电路逆序故障仿真实验结果扫描。第 10 届亚洲测试研讨会论文集，2001， 467.
[17] REDDY L N、POMERANZ I 和 REDDY S M. 罗特科：逆序测试压实技术。欧洲专用集成电路会议记录 '

92, 1992, 52242, 189–194。
[18] XIANG D、SUI W、YIN B 和 CHENG K. 用于发射捕获转换故障测试的具有影响输入测量的紧凑测试生成。IEEE 超大规模集成(超大规模集成电路)系统汇刊，2014，22(9)，1968–1979。
[19] NAVABI Z. 数字系统测试和可测试设计。马萨诸塞州波士顿：施普林格美国，2011。
[20] EL-MALEH A H、KHURSHEED S S 和 SAIT S M. 基于逆序恢复和测试松弛的时序电路的高效静态压缩技术。第 14 届亚洲测试研讨会论文集(自动变速器'05)，2005，378–385。
[21] DIMOPOULOS M 和 LINARDIS P. 通过应用集覆盖技术对测试序列集进行高效静态压缩。欧洲会议和展览的设计、自动化和测试论文集，2004，194–199。
[22] SAHARI M S、A'AIN A K 和 GROUT I。关于测试向量随机性对测试长度及其故障覆盖率影响的研究。第 10 届 IEEE 国际半导体电子会议(ICSE)论文集，2012，503–506。
[23] KHATRI A R、HAYEK A 和 BÖRSCÖK J. 使用建议的锉刀配合工具进行行为 Verilog 设计的故障注入和测试方法。国际高级计算机科学与应用杂志，2019, 10（4），57-63。
[24] KHATRI A R、HAYEK A 和 BÖRSCÖK J. 在建议的锉刀配合工具下为故障模型选择 SP 值的验证。第一届电气工程和计算技术最新趋势国际会议论文集(智力)，2017，1-7。
[25] KHATRI A R. 锉刀配合工具的技术指南。国际高级计算机科学与应用杂志，2019, 10(12), http://dx.doi.org/10.14569/IJACSA.2019.0101279
[26] MIRKHANI S 和 ABRAHAM J A. 使用基于模拟的统计指标快速评估测试向量集。IEEE 第 32 届超大规模集成电路测试研讨会(VTS)论文集，2014，1–6。
[27] HABIB K、SAFAR M、DESSOUKY M 和 SALEM A。在高考-ATPG 中不关心基于动态测试向量压缩。IEEE 第 57 届国际中西部电路与系统研讨会论文集，2014，213–217。
[28] DUNBAR C 和 NEPAL K。使用平台 FPGA 进行故障仿真和测试集生成以检测固定故障。计算机杂志，2011，6(11)，2335–2344。
[29] ZHANG Y. 卡住和转换故障的诊断测试模式生成和故障模拟。奥本大学，2012。