On Distrubution of Testability Values in Scan-Layout State-Space

Josef Strnadel
Faculty of Information Technology
Brno University of Technology, Božetěchova 2, 61266 Brno, Czech Republic
e-mail: strnadel@fit.vutbr.cz

Abstract

In the paper, it is shown how are testability values distributed within the scan-layout state-space for particular digital circuit. The goal of the paper was to approve or dismiss our hypothesis that the more registers are included in greater number of multiple scan-chains within particular scan-layout, the better testability properties correspond to the scan-layout.

Keywords: digital circuit diagnostics, register-transfer level, circuit data-path, testability analysis, design for testability, testability improvements, scan technique

1. Our previous work

1.1 Extension of transparency model

During our previous research, we have developed transparency-based model which is an extension of an I-Path based model of RTL circuit data-paths. Both theoretically and experimentally it was shown it is advantageous to analyze circuit data-path separately for transferring test vectors and responses using more general (so called virtual-port based) transparency concept [6, 7]. Using the concept, it is possible to use greater part of the circuit data-paths for diagnostic purposes in general, i.e., to utilize original in-circuit resources (data-paths, hardware resources, etc.) in a better way for diagnostic purposes. Main benefit of the concept can be seen in the possibility to model such parts of the circuit data-paths that are seen as unusable by previous techniques, even though they are usable to transport diagnostic data.

This work has been supported by Czech Grant Agency (GAČR) under contracts No. GA102/05/P193 *Optimizing Methods in Digital Systems Diagnosis* and No. GA102/04/0737 *Modern Methods of Digital Systems Design*

1.2 Testability analysis method

Also, we dealt with design and implementation of a *testability analysis* (TA) method for structurally described *register-transfer level* (RTL) circuits [6, 7].

Proposed TA algorithm is constructed as a graph-searching algorithm over G_S (test-vector data-flow digraph) and G_I (response data-flow digraph). During the search process, accessibility of virtual ports from circuit primary inputs is analyzed in G_S (this step corresponds to controllability analysis) and accessibility of virtual ports at circuit primary outputs is analyzed in G_I (this step corresponds to observability analysis).

Principle of a basic graph-searching (vertex-marking) algorithm is as follows. During searching process, marks are assigned to vertices within graph in such a way that if a mark is assigned to a vertex x a path from initial vertex r (i.e. from the only one marked vertex at the start) to vertex x exists in the graph. The principle of this basic algorithm is as follows.

1. [*Initialization*] Assign a mark to r; other vertices are markless.
2. [*Edge selection*] Select an edge e, whose start-vertex is marked and end-vertex markless; continue with step 3. If there

is no such edge, finish the algorithm.

3. [*Marking*] Assign a mark to the end-vertex of e, go to step 2.

Some of significant properties of proposed TA algorithm were prooved in [6] - especially, the time complexity $O(|V(G_S)| \cdot |E(G_S)| + |V(G_I)| \cdot |E(G_I)|)$ of the algorithm and correctness of the algorithm in view of testability-evaluating formulas. Because of their sizes, proofs are ommited in this text. Because the above-mentioned time complexity is the worst-case time complexity of the algorithm, it implies the algorithm is able to finish worst in quadratic time, achieving results comparable to other existing approaches to TA problem.

1.3 TA application

In order to verify theoretically proven properties of proposed TA algorithm, we have decided to apply the algorithm in two following areas:

- area dealing with automatic TA-driven generation of so-called RTL benchmark circuits (quality of generated benchmark circuits were succesfully proven by commercial test generation tools) [2]
- area dealing with automatic *design for testability* (DFT) process using scan technique [4, 5, 6, 7].

To be able to apply the algorithm in the second-mentioned area, i.e., to be able to determine which scan-layout[1] represents the most feasible alternative according to given design-constraints and desired diagnostic properties of resulting design, it was necessary to analyze the size of scan-layout state-space and mathematically describe particular scan-layout using special notation.

During our research activities (e.g., [4, 5, 6]), we have analyzed that for the circuit containing n registers (e.g., $n = 6$ in the circuit from Fig. 1) in its structure ($REG = \{R_1, \ldots, R_n\}$), it is possible to include $k \in \{1, \ldots, n\}$ of them into scan-chains (such registers are shadowed in the example in Fig. 1; additionally to parallel $n-bit$ input IN, $n-bit$ output OUT and clk port they are equipped with serial-in (out) port S_IN (S_OUT) and mode selection port -N/T) in $s1_n = \sum_{k=1}^{n}[\binom{n}{k} \times b_k]$ ways (in case sequence of registers within scan-chains is not important), where b_k is the number of partitions of a k-element set (so-called BELL number) or in $s2_n = \sum_{k=1}^{n}[\binom{n}{k} \times a_k]$ ways (in case sequence of registers within scan-chains is important), where a_k is so-called LAH number determining the number of partitions of a k-element list (ordered set) into i partition classes.

Also, formal description (notation) of a scan-layout was developed. Proposed notation [5, 4, 6] is related to the above-mentioned "partition analysis" of scan-layout search state-space. Let the notation be summarized in brief by following points:

1. scan-chain is represented by a sequence of registers chained within the chain,
2. special character (period, dot) is used to separate particular scan-chains,
3. if the ordering of scan registers within the scan-chains is not important, registers belonging to the same scan-chain are chained in the left-right direction according to increasing values of their indexes (registers with higher indexes are placed on the right),
4. scan-chains are ordered in a left-to-right way according to increasing index of the first register in a scan-chain (scan-chains with higher index of the first register are placed on the right),
5. if no register is selected into scan, notation contains only the special character.

As an example of scan-layouts corresponding to the notation, see Fig. 1. In Fig. 1a), R_1, R_2 and R_5 registers are included into scan, i.e. modified to scan registers. The registers are placed in two scan-chains in such

[1] particular way of selecting and chaining registers into scan-chains

a way the first scan-chain consists (in given order) of R_2 and R_1 registers and the second one consists of R_5 register. In Fig. 1b), registers R_1, R_2 and R_5 registers are included into scan again, but the first scan-chain consists of R_1 and R_2 registers in reverse order than in the previous case. In Fig. 1c), all registers are included into scan in a following way. Each of registers R_2, R_3, R_6 forms separate scan-chain and registers R_1, R_5 and R_4 form (in given order) further scan-chain. In total, there are two scan-chains in Fig. 1a) and 1b) and four scan-chains in Fig. 1c). As another example of scan-layouts described by the notation, let us present $R_1R_2.R_5$, $R_1R_4R_5.R_2.R_3.R_6$ where ordering of registers in scan-chain is not important and $R_2R_1.R_3$, $R_1R_5R_4.R_3.R_2.R_6$ where ordering of registers in scan-chain is important-registers are chained in the order as it is described in the notation.

2. Experimental results

To experimentally prove some of our TA-method properties, algorithm was implemented in C++ and applied on circuits from several benchmark sets (e.g. [1, 2, 3, 6]).

2.1 Testability analysis results

In the Fig. 2, relation between number of modules within the circuit structure and average time needed to perform TA is presented. The goal is to show our TA algorithm runs in linear-time in average (quadratic-time in the worst case). Numbers at the horizontal (vertical) axis represent thousands of modules present in the circuit structure (average time in seconds needed to perform TA of a circuit consisting of such a number of modules). The time is average because it differs little bit for various circuit structures consisting of the same number of modules.

In the chart it can be seen that till cca 1500 modules within the circuit structure, TA algorithm takes about one second or less of CPU time. The duration is almost constant because it takes longer time to construct dynamic data-structures for such "small" circuits than to analyze their testability. If there are more than cca 1500 modules within the circuit structure, TA overhead becomes greater than TA-startup overhead because of more complex circuit structures; CPU time can be seen as a linear function of the number of in-circuit modules in this case

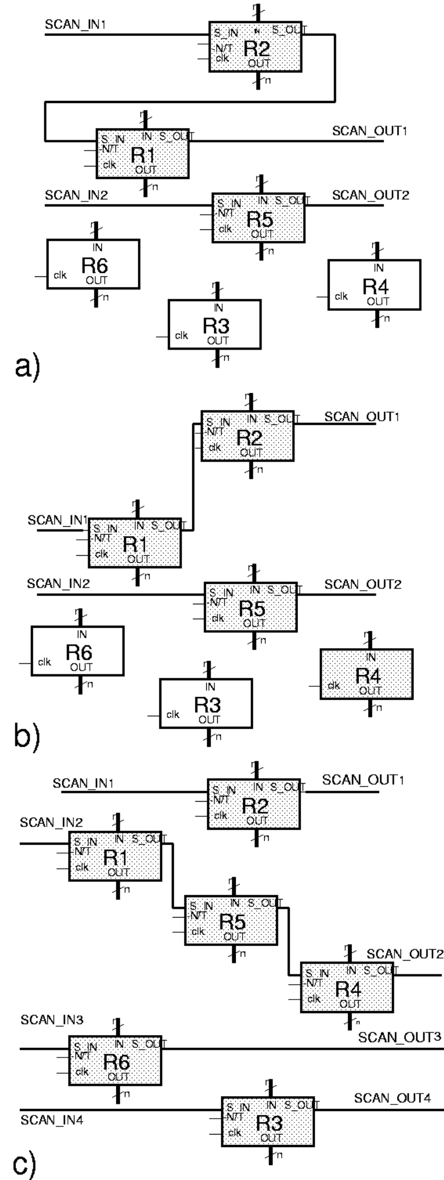

Fig. 1: Illustration to scan-layouts

Results presented in the chart were gained after application of our TA-method to a set of artificially generated circuits from [2] set.

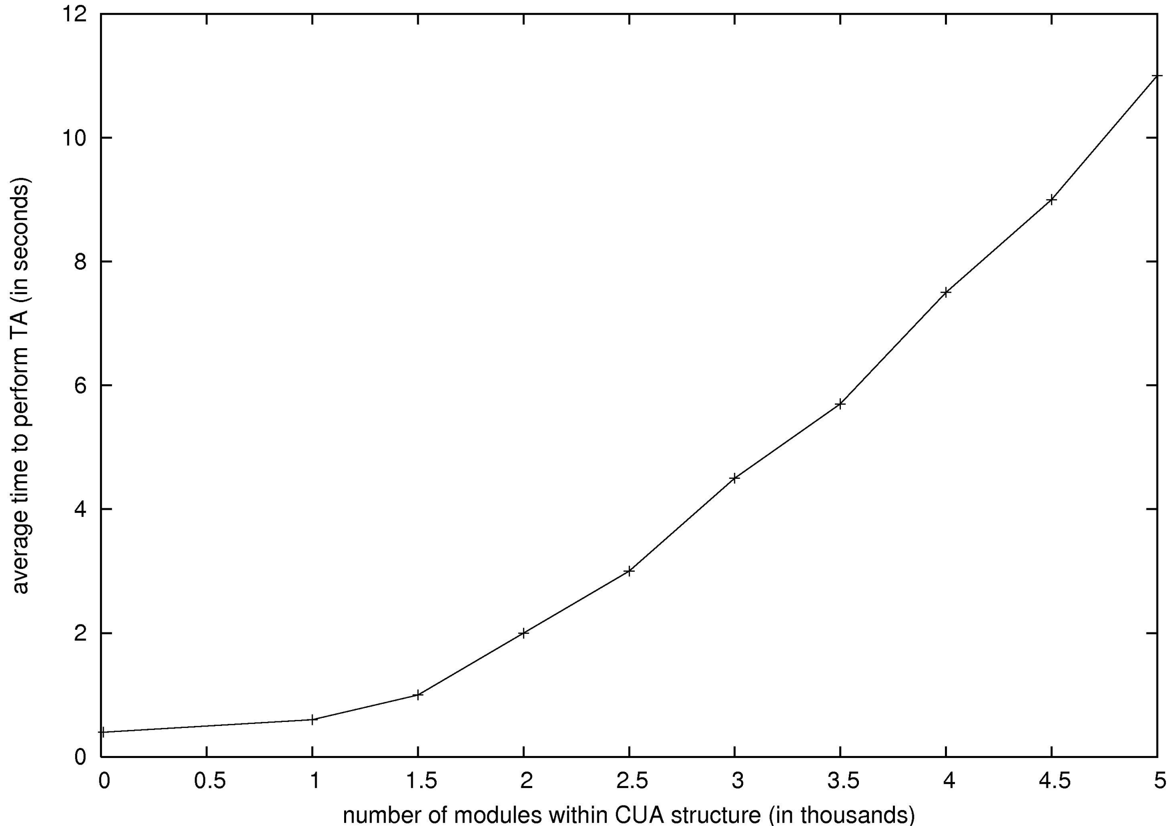

Fig. 2: Average CPU-time needed for proposed TA algorithm

2.2 Results of selecting registers into scan-chains

In the first-mentioned area (benchmark generation), our TA-method was used to find the worst-testable version of a circuit structure having desired structural properties. In the second area our TA-method was applied was a DFT area. The method was used here to find a solution for the problem of selecting registers into scan-chains. The goal was to use results provided by proposed TA method in the following way: it was required to select registers into scan-chains and to chain them in such a way that the highest possible testability of resulting design would be achieved together with the lowest price proportional to size of modifications caused by scan application in the original circuit structure.

Our TA-based method for selection of registers into scan-chains was implemented using several optimizing techniques. First, it was implemented using a genetic algorithm, then also using greedy-search algorithm and simulated annealing algorithm. As an input of the method, information about a circuit structure, set of DFT techniques allowed for testability improvement, diagnostic related requirements (e.g., desired fault coverage, test application time, number of testable nodes), design constraints (e.g., maximum area/pin overhead allowed) and selection of optimizing algorithm are used. At the output, a modification of original circuit structure fulfilling desired diagnostic properties and design constraints maximally appears.

3. Distrubution of testabilities in scan-layout state-space

During experiments done in the second-mentioned area, scan-layout has being searched that 1) meets given design-constraints maximally and 2) is characterized by maximal value of testability.

In connection with the experiments, we

Table 1: Exploration of scan-layout state-space for *Diffeq*: Summary

Pin overhead (%)		Area overhead (%) 8	10	14	16	20
5	scan-layout	R_6R_1	$R_6R_1R_3$	$R_6R_1R_3R_4$	$R_6R_1R_4R_2R_3$	$R_6R_1R_2R_4R_3$
	testability	0.631	0.658	0.678	0.689	0.7
10	scan-layout	R_6R_1	$R_6R_1R_2$	$R_6R_1R_4R_3$	$R_6R_1R_4R_2R_3$	$R_6R_1R_3R_1R_4$
	testability	0.631	0.658	0.678	0.689	0.747
15	scan-layout	$R_1.R_6$	$R_1.R_3.R_6$	$R_1R_4.R_6R_3$	$R_1R_2.R_6R_3R_4$	$R_1R_4R_2.R_6R_5R_3$
	testability	0.638	0.662	0.758	0.772	0.747
20	scan-layout	$R_1.R_6$	$R_1.R_2.R_6$	$R_1.R_3.R_6R_2$	$R_1R_3.R_2.R_6R_4$	$R_1R_5.R_3R_2.R_6R_4$
	testability	0.638	0.706	0.763	0.786	0.797
25	scan-layout	$R_1.R_6$	$R_1.R_3.R_6$	$R_1.R_2.R_3.R_6$	$R_1R_4.R_2.R_3.R_6$	$R_1R_4.R_2R_5.R_3.R_6$
	testability	0.638	0.706	0.748	0.784	0.798
30	scan-layout	$R_1.R_6$	$R_1.R_2.R_6$	$R_1.R_2.R_3.R_6$	$R_1.R_2.R_3.R_4.R_6$	$R_1.R_2R_4.R_3.R_5.R_6$
	testability	0.638	0.706	0.748	0.799	0.812
35	scan-layout	$R_1.R_6$	$R_1.R_2.R_6$	$R_1.R_2.R_3.R_6$	$R_1.R_2.R_3.R_4.R_6$	$R_1.R_2.R_3.R_4.R_5.R_6$
	testability	0.638	0.706	0.748	0.799	0.819

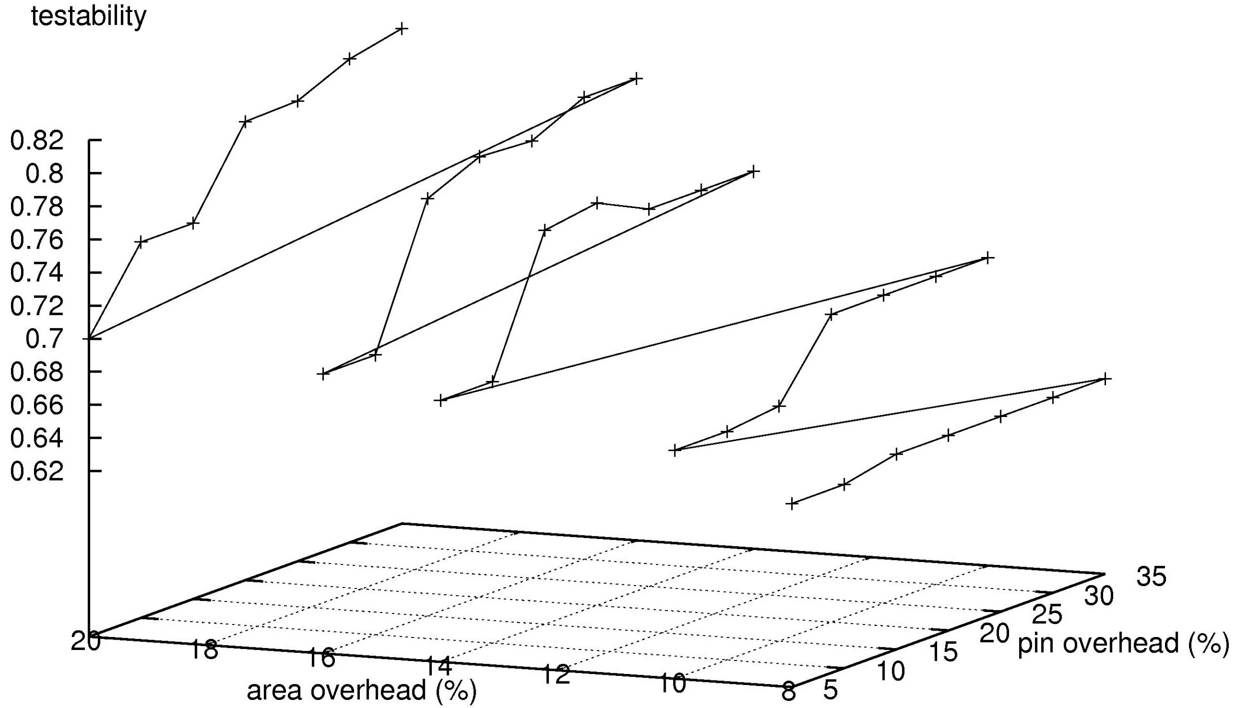

Fig. 3: Visualisation of testability distribution in *Diffeq* scan-layout state-space

were interested how testabilities are distributed within the scan-layout state-space for particular circuit. The goal was to approve or dismiss our hypothesis that the more registers are included in greater number of multiple scan-chains within particular scan-layout, the better testability properties correspond to the scan-layout.

By the experiments, our hypothesis has been approved. Experimental results gained for *Diffeq* benchmark circuit (described structurally in VHDL) are presented in Tab. 1 and Fig. 3. In Tab. 1, distribution of testabilities for particular scan-layouts (meeting pin and area overhead constraints – estimated by means of equivalent NAND gates – given in particular row and column of the table) are presented. The distribution is visualized in Fig. 3.

In the table, it can be seen:

1. the greater area overhead constraint given by user is, the more registers are included in scan-chains within the scan-layout,

2. the greater pin overhead constraint given by user is, the more multiple scan-chains are present in the scan-layout.

It can be seen that in case of low-overhead constraints, scan-layout contains less registers and less multiple schan-chains than in case of high-overhead constraints.

4. Conclusions

The main subject of our research was to deal with the TA of RTL digital circuit data-path and with utilizing its results in selected areas within the digital circuit diagnostic area. In the chaper 3 of the paper (Tab. 1 and Fig. 3), results of exploring *Diffeq* scan-layout state-space by our method are presented. The results show correctness of our TA measures as well as proposed TA algorithm and confirmed our hypothesis stated at the start of chapter 3 is right.

In the future, we plan to extend our model about more DFT techniques and to perform new set of experiments able to find solutions with better cost/quality trade-off ratio. Also, we plan to deal with analysis of correlation between our TA measures and fault-coverage parameter provided by academic/commercial test generation tools.

References

[1] Bukovjan, P.: *Allocation for Testability in High-Level Synthesis*. PhD thesis, Institute National Polytechnique de Grenoble, 2000. 130 p.

[2] Pečenka, T.—Kotásek, Z.—Sekanina, L.,—Strnadel, J.: Automatic Discovery of RTL Benchmark Circuits with Predefined Testability Properties. In *Proceedings of the 2005 NASA/DoD Conference on Evolvable Hardware*, IEEE Computer Society Press, 2005. pp. 51–58.

[3] Růžička, R.: *Formal approach to the Testability Analysis of RT Level Digital Circuits* (in Czech). PhD thesis, VUT v Brně, 2002. 102 p.

[4] Kotásek, Z.—Strnadel, J. Optimising Solution of the Scan Problem at RT Level Based on a Genetic Algorithm. In *Proceedings of 5th IEEE Workshop on Design and Diagnostics of Electronics Circuits and Systems*, FIT VUT v Brne, 2002. pp. 44–51.

[5] Kotásek, Z.—Strnadel, J.: Testability Improvements Based on the Combination of Analytical and Evolutionary Approaches at RT Level. In *Proceedings of Euromicro Symposium on Digital System Design - Architectures, Methods and Tools*, IEEE Computer Society Press, 2002. pp. 166–173.

[6] Strnadel, J.: *Testability Analysis and Improvements of Register-Transfer Level Digital Circuits* (in Czech). PhD thesis, VUT v Brně, 2004. 150 p.

[7] Strnadel, J.: VIRTA: Virtual Port Based Register-Transfer Level Testability Analysis and Improvements. In *Proceedings of 8th IEEE Design and Diagnostic of Electronic Circuits and Systems Workshop*, University of West Hungary, 2005. pp. 190–193.