NESTED LOOPS DEGREE IMPACT ON RTL DIGITAL CIRCUIT TESTABILITY[1]

Josef Strnadel[2]

Brno University of Technology,
Faculty of Information Technology,
Božetěchova 2, 612 66 Brno, Czech Republic

Abstract: The existence of loops in a circuit structure causes problems in both test generation and application. Thus, the problem of identifying loops becomes an important task during testability analysis or, later, e.g., during allocation-for testability process. When nested loops occur in the circuit, it is necessary to accurately determine the most nested one to improve circuit testability significantly, with minimal design cost. This paper deals with the problem of identifying nested loops including their nesting degree in the register-transfer level (RTL) digital circuit structure as well as with the impact of such loops on the circuit testability. *Copyright © 2003 IFAC*

Keywords: Algorithms, Digital circuits, Diagnostic, Design, Directed graphs, Feedback loops, Testability.

1. INTRODUCTION

As digital circuits (DCs) became larger and more complex, testing of electronic devices at both chip and board level has been more and more difficult and has become a large part of a DC device cost. While for simple-structure combinational DCs a test (including test vectors, structures, methods etc.) can be designed manually and especially for this circuit, this is almost impossible for complex sequential DCs. So, certain parts related to a DC testing problem are to be automated.

Digital circuit ability to be easy and efficiently tested–it means with a short test sequence length and high fault coverage–is called a *testability* of a digital circuit. Alike, we can talk about a *port (node), link testability* etc. Let us note that techniques leading to a design of high-testable circuits are called *design for testability* (*DFT*) techniques (intuitive and empiric techniques, structural design techniques, automated synthesis etc.). It can be said that DFT refers to hardware design styles or an added hardware that reduces test generation complexity.

Also, so called *synthesis for testability* (*SFT*) approaches exist that are able to relocate the testability improvements from the DC-design phase to the DC-synthesis phase. In the paper, we will refer only to DFT, but not to SFT.

Usually, usage of DFT technique(s) is preceded by a process called *testability analysis* (TA). For illustration of TA goals, let us specialize on *node testability* only in the paper. Due to this premise, the task of a TA process is to evaluate each *CUA* (circuit under analysis) node by so called *testability measures* [e.g., Strnadel, 2002a], which means assign a numeric value expressing a testability level of this node to it. In our [Strnadel, 2002a] approach, the more a node is testable, the higher numeric value is assigned to it.

Various approaches to testability analysis build on various principles exist. The best known is SCOAP [Goldstein, 1989] approach and from the last ones name [Bukovjan, 2000] approach or [Xinli, 1996] incremental testability analysis approach. Usually, a testability analysis process involves a static topological analysis of a digital circuit structure, but no test vectors and no search algorithm. However,

[1] The research is original, not published yet. It is supported by the Czech national grant GACR102/01/1531.

[2] Contact: WWW (http://www.fit.vutbr.cz/~strnadel), E-mail (strnadel@fit.vutbr.cz), Tel. (+420 541141208).

these lacks can be fixed either by a cooperation of a static testability analysis tool and a test sequence identification tool or using a dynamic testability approach, e.g. [Mao, *et al*, 1988]. Results of our previous research in this area are presented in [Kotasek, *et al.,* 2001a; Kotasek, *et al.,* 2001b; Hlavicka, *et al.,* 2001].

Thus, the main task of a TA as seen in the paper is to evaluate all circuit nodes by testability measures to be able to locate low-testability nodes. Then, DFT process can be started to provide a modification of a circuit structure leading to testability enhancement of these low-testability nodes. Consequently (with a very high probability), a testability enhancement of other nodes follows. Thus, the main goal of DFT process is to modify original circuit structure minimally but improve testability of most circuit nodes maximally. Essentially, the process of searching a trade-off between these two requirements is an iterative process (e.g., see [Strnadel, 2002b]).

On the basis of TA results only, it is not always possible to say how it is feasible to modify CUA structure to obtain maximal testability improvement at the acceptable costs of final design. Hence, it is suitable to complete TA approaches with methods investigating structural properties of CUA in detail. E.g., it is possible to show that feedback loops belongs to structures, which affect CUA diagnostic properties most negatively. To be low-testability nodes (i.e., difficult-to-test-nodes) detected most accurately to be the best testability improvement possible achieved, it is necessary to deal also with nested loops analysis and loop dependency analysis– especially with the loop nesting degree analysis.

In this paper, we will deal with the 1) description of loop detection method, 2) loop dependency analysis and 3) demonstration of nested loop impact to CUA testability.

2. SEARCHING LOOPS IN A DIGRAPH

Let $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V} = \boldsymbol{PORT_{CUA}}$ is a vertex set and $\boldsymbol{E} \subseteq \boldsymbol{V} \times \boldsymbol{V}$ is an edge set $((x, y) \in \boldsymbol{E} \Leftrightarrow (x, y) \in \boldsymbol{IP})$ be a digraph representing the structure of all *I paths* existing within CUA. Each edge is oriented in the means of data flow in the CUA data path. Two CUA ports $x, y \in \boldsymbol{PORT_{CUA}}$ are in $\boldsymbol{IP}$ relation iff it is possible to transfer data unchanged from port x to port y. Usually, if $(x, y) \in \boldsymbol{IP}$, bit-width of x is the same as the bit-width of y; but also, a special case of "partial" unchanged data transfer from x to y can be described using digital circuit model [Strnadel, 2002a].

Except graphical representation of a digraph, there are several other ways a digraph can be represented – adjacency list, adjacency matrix, incidence matrix etc. The major advantage of matrix representation is that the calculation of paths and cycles/circles can easily be performed using well-known operations of matrices. However, the disadvantage is that this form of representation takes away from the visual aspect of graphs. It would be difficult to illustrate in a matrix, properties that are easily illustrated graphically.

Let $\boldsymbol{A}$ be an incidence matrix of $\boldsymbol{G}$. By numbering of $\boldsymbol{G}$ edges in an appropriate way, $\boldsymbol{A}$ is possible to be in the form

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix}, \qquad (1)$$

where $\boldsymbol{A}_{12}$ is a regular square matrix of the $(|\boldsymbol{V}| - 1)^{\text{th}}$ order. It can be shown that $\boldsymbol{A}_{12}$ contains edges (columns) that are edges of certain spanning tree of $\boldsymbol{G}$. Then

$$\boldsymbol{B}_{\text{f}} = [\boldsymbol{E}, \boldsymbol{A}_{11}^{\text{T}} \times (\boldsymbol{A}_{12}^{-1})^{\text{T}}] \qquad (2)$$

is a matrix of fundamental system of circles incident to $\boldsymbol{A}_{12}$ ($\boldsymbol{A}_{12}$ is a spanning tree of $\boldsymbol{G}$), i.e., it is a fundamental system of feedback loops in CUA. Making linear combinations of $\boldsymbol{B}_{\text{f}}$ rows, other CUA circles are revealed and matrix $\boldsymbol{B}$ of all CUA circles is constructed.

3. LOOP DEPENDENCES

For digraph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ belonging to certain CUA, let $\boldsymbol{E} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_n\}$ be an edge set of $\boldsymbol{G}$ and let $\boldsymbol{K}_1, \boldsymbol{K}_2, ..., \boldsymbol{K}_m$ be all circles of $\boldsymbol{G}$ (i.e., all CUA loops). Then $\boldsymbol{B} = [\boldsymbol{b}_{ij}]$ is a matrix of type (m, n), where i^{th} row belongs to circle $\boldsymbol{K}_{\text{i}}$ and identifies whether edge $\boldsymbol{e}_j$ is $(\boldsymbol{b}_{ij} = 0)$ or is not $(\boldsymbol{b}_{ij} = 1)$ edge of $\boldsymbol{K}_{\text{i}}$.

Having all circles stored in $\boldsymbol{B}$ matrix, it is important to detect, which of them have the most impact on CUA testability. Generally, it is needed to analyze the dependence of CUA loops, i.e. whether loops are independent (sets of their edges are disjoint), or whether loops are dependent in such a way one is nested to the other one. We do not deal with the case of independent loops, because they are uncomparable from our point of view. But, we deal with dependent loops that are comparable, i.e., for two comparable loops it is possible to state which of them is the nested one.

For the purposes of comparing CUA loops, poset over the set of all CUA loops is defined. Let $\boldsymbol{M}$ be a set of all $\boldsymbol{B}$ rows (i.e., each element of $\boldsymbol{M}$ represents certain CUA loop) and let $(\boldsymbol{M}, <)$ be a sharp poset defined as follows:

If $(x \text{ bitAND } y) \neq \vec{0}$ then

i. x, y are comparable

ii. $x < y \Leftrightarrow [(x \text{ bitOR } y) = y]$

for any $x, y \in \boldsymbol{M}$; $x \neq y$.

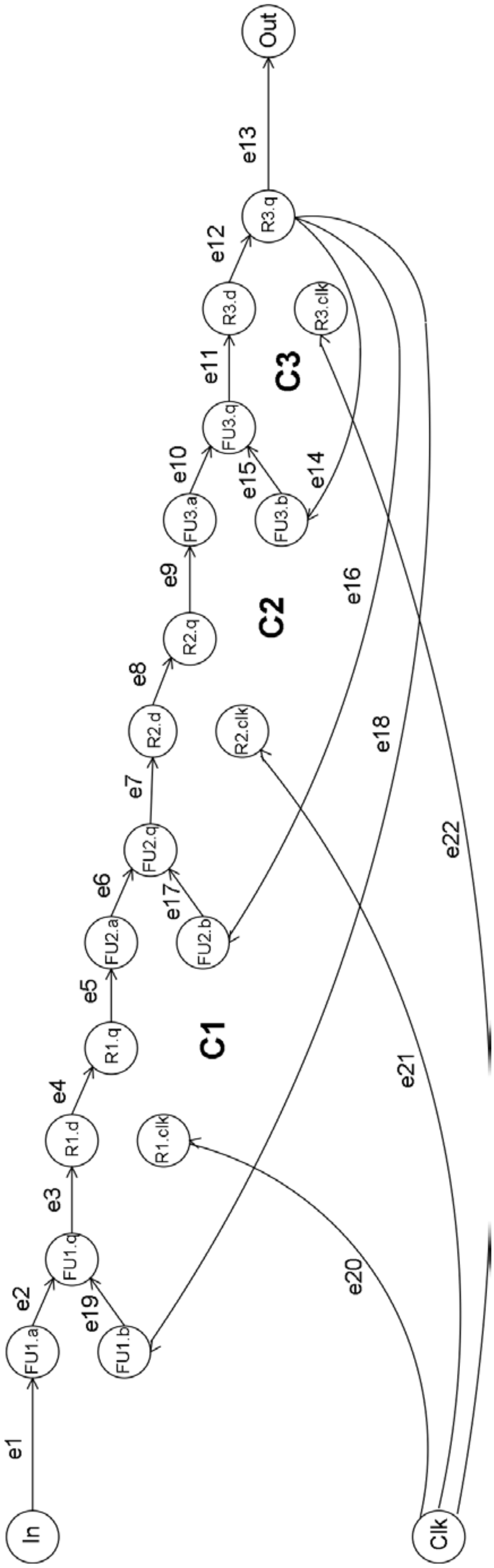

Fig. 1. Digraph (***G***) of NLoopsA circuit

We are interested in all minimal elements of (***M***, <) for given CUA, because they represent the most nested loops, which are most suitable to be broken by selected DFT technique(s) for improvement of CUA testability maximally and with minimal costs.

4. ILLUSTRATION

Let the example illustrating terms from previous two sections be presented in this section. Let it be demonstrated for NLoopsA circuit from Fig. 4.

Form of matrix ***B*** for NLoopsA is $\begin{array}{c|ccc|} & e1 & \ldots & e22 \\ \hline C1 & & & \\ C2 & & & \\ C3 & & & \\ \hline \end{array}$

$$\boldsymbol{B} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 2. Matrix of circles (***B***) for NLoopsA

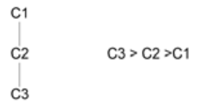

Fig. 3. Hasse diagram of (***M***, <) poset

5. EXPERIMENTAL RESULTS

In this section, impact of nested feedback loops on digital circuit testability will be demonstrated on concrete simple digital circuits as well as the impact of breaking the most nested loop by insertion of an extra multiplexer.

First of all, let the experimental DC be presented– NLoopsA. It is easy to discover that there are 3 loops in NLoopsA (C1, C2 and C3–see Fig. 1). They are nested in such a way that C2 is nested into C1 and C3 is nested into C2.

In Table 1 and Table 3, it can be seen that due to the system of nested loops existing in NLoopsA, the testability of NLoopsA is very low.

In Table 3, it can be seen there are only 28.6 % of controllable nodes (C-nodes) and 19 % of observable nodes (O-nodes) and there is no node (T-node) that is both controllable and observable in NLoopsA.

In Table 1, controllability, observability and testability values of each NLoopsA node is presented. The more is value close to 1, the easier is a node accessible using primary pins. Again, the more is value close to 0, the more difficult a node is accessible using primary pins. The boundary value 1 is assigned only to a node that is a primary pin or that is directly connected to a primary pin. Again, the boundary value 0 is assigned only to a node that is not anyway accessible using primary pins.

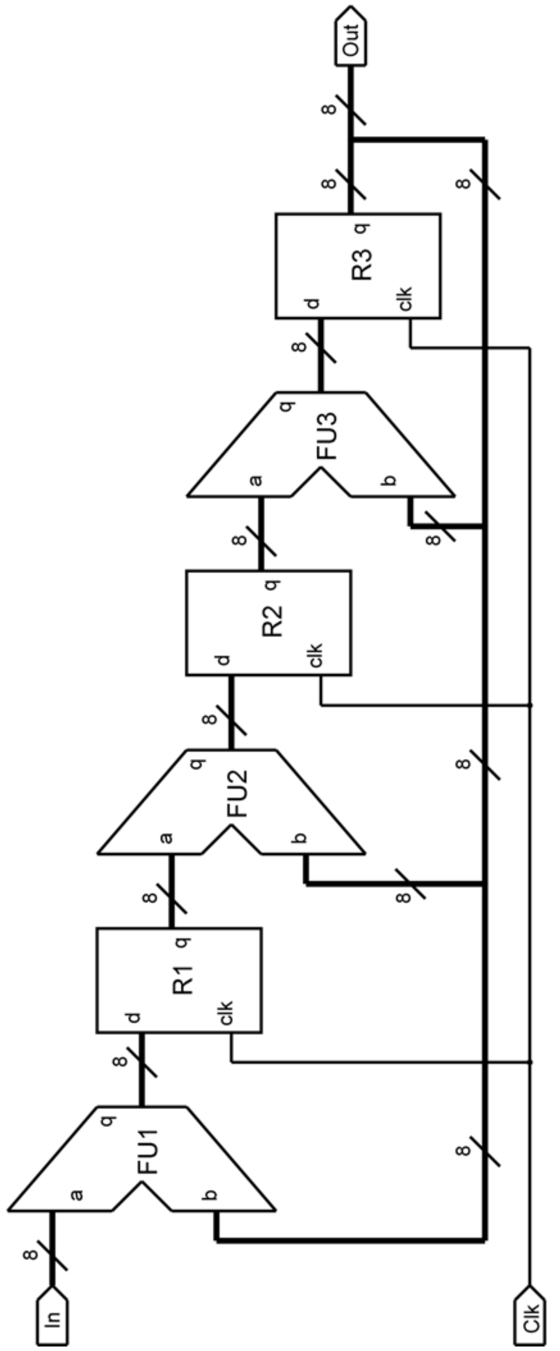

Fig. 4. NLoopsA circuit schema - nested loops make circuit almost untestable

Applying procedures presented in section 3 and section 4 to NLoopsA circuit, it is enumerated there is only 1 element (C3) in a set of all minimal elements of (***M***, <) poset–see Fig. 3. This information is very important, because consequently, it means that not by breaking of C1 or C2 loop or their combination, but by breaking of C3 loop, NLoopsA testability will improve significantly. This fact will be documented in the next text and tables. Of course, C1 and C2 can be broken in addition with C3–this can result to yet another (but not significant) testability improvement at the extra costs.

There are several ways of how to break C3 loop. One possibility is to modify R3 to scan register. In this case, C3 loop is broken by a serial data path provided by a scan register in a test mode of operation.

Table 1 Testability analysis results demonstrating nested loops impact on circuit testability

Port name	Con.	Obs.	Tst.
R3.q	0	1	0.5
R3.clk	1	0	0.5
R3.d	0	0.692	0.346
R2.q	0	0	0
R2.clk	1	0	0.5
R2.d	0	0	0
R1.q	0	0	0
R1.clk	1	0	0.5
R1.d	0	0	0
Fu3.q	0	0.692	0.346
Fu3.b	0	0	0
Fu3.a	0	0	0
Fu2.q	0	0	0
Fu2.b	0	0	0
Fu2.a	0	0	0
Fu1.q	0	0	0.5
Fu1.b	0	0	0.5
Fu1.a	1	0	0.5
NLoopsA.Out	0	1	0.5
NLoopsA.Clk	1	0	0.5
NLoopsA.In	1	0	0.5

Table 2 Testability analysis results after breaking the most nested loop

Port name	Con.	Obs.	Tst.
MX1.q	0.937	0.708	0.823
MX1.sel	1	0	0.5
MX1.b	1	0.666	0.833
MX1.a	0.098	0.666	0.382
R3.q	0.656	1	0.828
R3.clk	1	0	0.5
R3.d	0.937	0.708	0.823
R2.q	0.161	0.410	0.286
R2.clk	1	0	0.5
R2.d	0.262	0.388	0.325
R1.q	0.430	0.236	0.333
R1.clk	1	0	0.5
R1.d	0.615	0.143	0.379
Fu3.q	0.098	0.666	0.382
Fu3.b	0.656	1	0.828
Fu3.a	0.161	0.410	0.286
Fu2.q	0.262	0.388	0.325
Fu2.b	0.656	1	0.828
Fu2.a	0.430	0.236	0.333
Fu1.q	0.615	0.143	0.379
Fu1.b	0.656	1	0.828
Fu1.a	1	0.042	0.521
NLoopsA.Out	0.656	1	0.828
NLoopsA.Clk	1	0	0.5
NLoopsA.Sel	1	0	0.5
NLoopsA.Dx	1	0.666	0.833
NLoopsA.In	1	0.042	0.521

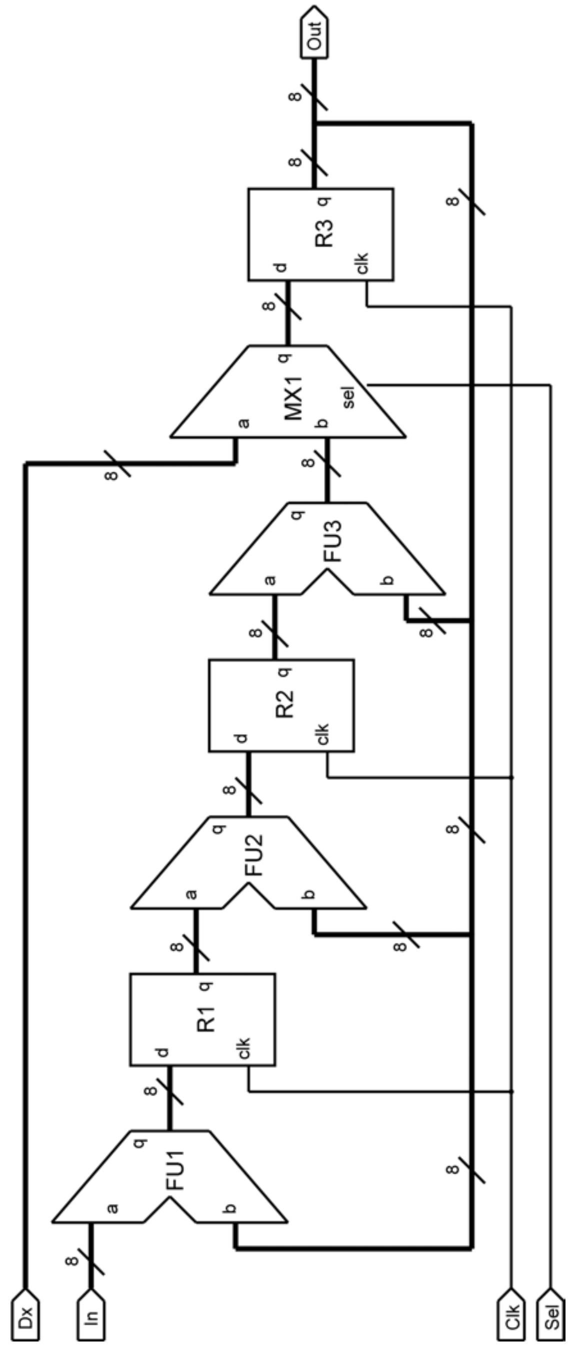

Fig. 5. NLoopsB circuit schema, i.e., NLoopsA schema with most nested loop broken by an extra multiplexer in circuit data-path

Next, C3 loop can be partially broken by adding init mode of operation to FU3, by inserting some BIST element into the loop or using any other technique.

In our case (see Fig. 5), it was decided to break C3 loop by inserting an extra multiplexer MX1 into a NLoopsA data path (modified NLoopsA circuit is denoted NLoopsB). This was done at the costs of a certain NLoopsA area, delay and pin overhead growth. Together with MX1 structure (that represents certain area overhead and extra delay in circuit data path), primary inputs Dx and Sel were added into NLoopsA interface. If Sel=0, data from FU3.q are loaded to R3.d port, as in the case of unmodified NLoopsA. If Sel=1, desired test data from Dx primary input are loaded to R3.d.

In Table 2 and Table 3, testability parameters of NLoopsB circuit are presented. In both tables, it can be seen that testability of modified NLoopsA circuit improved significantly, compared with testability parameters of original NloopA circuit.

In Table 3, it can be seen there are 100 % of C-nodes (it is 71.4 % more than in NLoopsA case), 71.8 % of O-nodes (it is 52.8 % more than in NLoopsA case) and there 71.8 % of T-nodes (it is 71.8 % more than in NLoopsA case) in NLoopsB. In NLoopsB, the only nodes that are not observable and testable are clock and select input ports. This is because limitations of testability analysis used [Strnadel, 2002a].

Table 3 NLoopsA/B properties

Circuit	C-nodes	O-nodes	T-nodes
NLoopsA	6 [28.6 %]	4 [19 %]	0 [0 %]
NLoopsB	27 [100 %]	21 [71.8 %]	21 [71.8 %]

In Table 2, it is seen that because of breaking C3 loop, the output q of R3 become controllable. Consequently, b ports of all FUs become controllable also, with the same controllability value, because they are directly connected to R3.q. Next, controllability of all other nodes has improved.

Similar testability parameters would be achieved when C3 loop is broken using any other DFT technique. In dependence of DFT technique applied (certain delay, pin, area overhead etc. is related to a DFT technique), the properties of modified NLoopsA circuit would be very close to the values (achieved by using multiplexer insertion technique) presented in column 2 of Table 3. But, more detail results as are presented in Table 2 could differ more significantly with respect to DFT technique used.

6. CONCLUSIONS

In the paper, the relation between testability and nested loops occurrence was presented. It was shown that dealing with feedback loop analysis deeply makes sense, because it increases accuracy of locating difficult-to-test nodes; on the basis of analysis result, it is possible to improve circuit testability significantly, with minimal design cost. Also, impact of nested feedback loops on digital circuit testability was demonstrated on concrete simple digital circuit as well as the impact of breaking the most nested loop by insertion of an extra multiplexer.

7. ACKNOWLEDGEMENTS

This work has been financially supported by the Grant Agency of Czech Republic grant No. 102/01/1531 “Formal Approaches in Digital Circuit Diagnostics – Testable Design Verification”

REFERENCES

Abadir M. S., Breuer M. A. (1985): A Knowledge Based System for Designing Testable VLSI chips, *IEEE Design&Test*, August1985, pp. 56-68

Bukovjan P. (2000): Allocation for Testability in High-Level Synthesis, *PhD thesis*, Institut National Polytechnique de Grenoble.

Goldstein, L. H.: SCOAP (1980): Sandia Controllability/Observability Analysis Program, In: *Proceedings of 17th Design Automation Conference*, Minessota, June 1980

Hlavička J., Kotásek Z., Růžička R., Strnadel J. (2001): Interactive Tool for Behavioral Level Testability Analysis, In: *Proceedings of the IEEE ETW 2001*, Stockholm, pp. 117-119

Kotásek Z., Růžička R., Strnadel J., Zbořil F. (2001): Two Level Testability System, In: *Proceedings of the 35th Spring International Conference MOSIS'01*, Ostrava, MARQ, pp. 433-440

Kotásek Z., Růžička R., Strnadel J. (2001): Formal and Analytical Approaches to the Testability Analysis - the Comparison, In: *Proceedings of IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop 2001*, Györ, pp. 123-128

Mao W., Ciletti M. D. (1988): Dytest: A Self-Learning Algorithm Using Dynamic Testability Measures to Accelerate Test Generation, In: *Proceedings of the 25th ACM/IEEE Conference on Design Automation*, DAC '88, Anaheim, CA, pp. 591-596

Strnadel J. (2002): Normalized Testability Measures Based on RTL Digital Circuit Graph Model Analysis, In: *Proceedings of The fifth International Scientific Conference Electronic Computers and Informatics 2002*, Košice, TU v Košiciach, pp. 200-205

Strnadel J., Kotásek Z. (2002): 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 DSD'2002*, Los Alamitos, US, ICSP, pp. 166-173

Xinli G. (1996): RT Level Testability Improvement by Testability Analysis and Improvements, *PhD thesis*, Department of Computer and Information Science, Linköping University, Sweden