Normalized Testability Measures Based on RTL Digital Circuit Graph Model Analysis

Josef Strnadel
Department of Computer Systems, Faculty of Information Technology,
Brno University of Technology, Božetěchova 2, 612 66 Brno, Czech Republic
E-mail: strnadel@fit.vutbr.cz

Abstract

The paper presents measures for testability valuation of a digital circuit at register-transfer level (RT level, RTL). Definition of a graph model of a RTL digital circuit for these purposes and mathematical formulas of measures for testability valuation based on analysis of proposed graph model are presented in this paper. Finally, experimental results are presented.

1. Introduction

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

Digital circuit ability to be easy and efficiently tested – it means with a short test sequence length and high fault coverage – is called *testability* of a digital circuit. Alike, we can talk about *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.

Usually, usage of DFT technique(s) is preceded by a process called *testability analysis*. For illustration of testability analysis goals, let us specialize on *node testability* only in this preface section. Due to this premise, the task of a testability analysis process is to evaluate each circuit node by so called *testability measures*, which means assign a numeric value expressing testability level of this node to it. In our approach, the more a node is testable, the higher numeric (real number) value from <0; 1> interval is assigned to it. Various approaches to testability analysis build on various principles exist. The best known is SCOAP [1] approach and from the last ones name [2] approach or [3] 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, 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. [4]. Results of our previous research in this area are presented in [5], [6] or [7].

Thus, the main task of a testability analysis 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, a testability enhancement of other nodes usually 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, which is topic of another research.

As stated above, many approaches to testability analysis exist. In our research, we specialize in development of an automated DFT process. In addition to DFT process method (usually some iterative process), we need 1) a circuit model representing circuit structure and 2) a function for evaluating each partial solution (modified original circuit structure) of the DFT process by a numeric value commensurably to solution quality with a respect to user requirements (max. chip area overhead, max. chip pin overhead, min. chip testability, max. test application time etc.). Thus, one of the aspects this function is based on is testability of given circuit; its fundamentals in the form of testability measures are presented in this paper.

The paper is organized as follows: First, the proposed model of RTL digital circuit, and testability metrics built over this model are presented. Finally, the experimental results and perspectives of future research activities are introduced in brief.

2. Graph Model

Let a RTL digital circuit be denoted ***UUA*** (Unit Under Analysis). In this section, only basic building blocks of our model will be presented to imagine what is our model build on. The reason is there is not enough space for presenting whole circuit model in this paper.

Let $\boldsymbol{ITM_{UUA}} = \boldsymbol{REG_{UUA}} \cup \boldsymbol{FU_{UUA}} \cup \boldsymbol{MUX_{UUA}} \cup \{\boldsymbol{cir_{UUA}}\}$ be a set of all ***UUA*** items, where $\boldsymbol{REG_{UUA}}$ is a set of ***UUA*** registers, $\boldsymbol{FU_{UUA}}$ is a set of ***UUA*** functional units, $\boldsymbol{MUX_{UUA}}$ is a set of ***UUA*** multiplexers and $\boldsymbol{cir_{UUA}}$ is a special element identifying whole ***UUA*** circuit.

Each circuit item from $\boldsymbol{ITM_{UUA}}$ set has its interface consisting of input and output ports. ***UUA*** ports are (in respect to their type) sectioned into several sets and each port belongs exactly to one set. Let $\boldsymbol{PORT_{UUA}} = \boldsymbol{PIN_{UUA}} \cup \boldsymbol{POUT_{UUA}} \cup \boldsymbol{IN_{UUA}} \cup \boldsymbol{CIN_{UUA}} \cup \boldsymbol{OUT_{UUA}}$ be a set of all ***UUA*** ports, where $\boldsymbol{PIN_{UUA}}$ is a set of all ***UUA*** primary inputs, $\boldsymbol{POUT_{UUA}}$ is a set of all ***UUA*** primary outputs, $\boldsymbol{IN_{UUA}}$ is a set of input ports of all inner ***UUA*** items (inner ***UUA*** item is from $\boldsymbol{ITM_{UUA}}\backslash\{\boldsymbol{cir_{UUA}}\}$ set), $\boldsymbol{CIN_{UUA}}$ is a set of input control/synchronization ports of all inner ***UUA*** items, $\boldsymbol{OUT_{UUA}}$ is a set of output ports of all inner ***UUA*** items.

In an ***UUA*** circuit structure, inner circuit items are interconnected using their interface ports. So a definition of a link is also needed in our model. Let $\boldsymbol{L} \subset (\boldsymbol{PIN_{UUA}} \cup \boldsymbol{OUT_{UUA}}) \times (\boldsymbol{POUT_{UUA}} \cup \boldsymbol{IN_{UUA}} \cup \boldsymbol{CIN_{UUA}})$ be a relation representing a link connecting two ***UUA*** ports – two ports are in $\boldsymbol{L}$ relation, when a link connecting them exists in the ***UUA*** structure.

For the purposes of diagnostic data transfers through ***UUA*** structure there was also a need of modeling terms presented in [8]: an ***UUA*** item $itm \in \boldsymbol{ITM_{UUA}}$ with an input port x and an output port y is said to have an *identity mode (i mode),* if *itm* has a mode of operation in which the data from port x can be transferred unchanged to port y. A path from x to y fulfilling these properties is called *i path* and is built into our model in the form of following relation.

Let $\boldsymbol{I} \subset \boldsymbol{IN_{UUA}} \times \boldsymbol{OUT_{UUA}}$ be a relation representing inner i path through an ***UUA*** item with transparency mode (i mode) – two ports x, y of the same item are in $\boldsymbol{I}$ relation, when it is possible to transport data unchanged from x port to y port.

Having information about connection(s) among ***UUA*** ports and transparency mode(s) of each *itm* from $\boldsymbol{ITM_{UUA}}$ set, (global) circuit i path can be represented in our model as follows.

Let $\boldsymbol{IP} \subset \boldsymbol{PORT_{UUA}} \times \boldsymbol{PORT_{UUA}}$ be a relation representing a circuit i path – two ports x, y are in $\boldsymbol{IP}$ relation, when $(x, y) \in \boldsymbol{L}$ or $(x, y) \in \boldsymbol{I}$. $(x, y) \in \boldsymbol{IP} \Leftrightarrow \overline{\boldsymbol{IP}}_x^y$ i path exists.

Even thought only a simplified version of our model was presented in previous paragraphs, a graph of ***UUA*** i paths definition can be sketched here.

Let $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ be a directed graph, where $\mathbf{V} = \boldsymbol{PORT_{UUA}}$ is a vertex set and $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$ is an edge set, where $(x, y) \in \mathbf{E} \Leftrightarrow (x, y) \in \boldsymbol{IP}$.

Let us deal with proposed testability metrics in the next section now.

3. Testability Analysis

Before presenting mathematical formulas for testability measures, basic demands on the measures will follow. So, let us specialize on claims posed on testability values. These claims led to the current form of proposed mathematical formulas for testability measures. Let us also note that there are two basic components of testability – one component is called *controllability*, the other one is called *observability*. The controllability one represents a node (port from $\boldsymbol{PORT}_{\boldsymbol{UUA}}$ set) ability to be easy and efficiently set up from circuit primary input ports. Similarly, the observability one represents a node ability to be easy and efficiently observed at circuit primary output ports. Getting these two factors together, testability of the node can be evaluated. Controllability, observability and testability values are supposed to be real numbers from <0; 1> interval.

First claim is posed on controllability values. It is requested that only a node uncontrollable from circuit primary input ports has a controllability value equal to zero. It means if there is no way how to get a certain value from circuit primary input ports to a given node, controllability of this node will be set to zero. Alike, it is requested that only a primary input port or a node connected to a primary input port has a controllability equal to one, because there is no difficulty of setting a value to this node except setting this value to a circuit primary input port. Controllability values of other nodes are requested to be from (0; 1) open interval.

Next claim is posed on observability values. It is requested that only a node unobservable at circuit primary output ports has an observability value equal to zero. It means if there is no way how to observe a certain value from a given node at circuit primary output ports, observability of this node will be set to zero. Alike, it is requested that only a primary output port or a node connected to a primary output port has an observability equal to one, because there is no difficulty of observing a value of this node except observing this value at a circuit primary output port. Observability values of other nodes are requested to be from (0; 1) open interval.

Last claim is posed on testability values. It is requested that only a node with both controllability and observability equal to zero has also testability equal to zero, because there is no way of testing this node using circuit primary ports. Similarly, it is requested that only a node with both controllability and observability equal to one has also testability equal to one, because there is no difficulty of testing this node except accessing circuit primary ports. Testability values of other nodes are requested to be from (0; 1) open interval.

After a presentation of basic requirements on controllability, observability and testability values, mathematical formulas fulfilling these claims can be presented.

Suppose $\overline{\boldsymbol{IP}}_{\boldsymbol{UUA}}$ is a set of all circuit i paths (for detail definition see [8]) existing in ***UUA***.

Let $ip \in \overline{\boldsymbol{IP}}_{\boldsymbol{UUA}}$. Suppose mappings $\iota_{IPAS}(ip)$, $\iota_{\#CLKS}(ip)$, $\iota_{\#PORTS}(ip)$, $\nu_{\#CLKS}(ip)$, $\nu_{\#PORTS}(ip)$ are defined, where $\iota_{IPAS}(ip)$ is a mapping assigning „a set of all ports that must be set up for adjusting *ip*“ to *ip*, $\iota_{\#CLKS}(ip)$ is a mapping assigning „a number of clk cycles that must be generated for adjusting *ip*“ to *ip*, $\iota_{\#PORTS}(ip)$ is a mapping assigning „a number of ports that must be set up for adjusting *ip*“ to *ip*, $\nu_{\#CLKS}(ip)$ is a mapping assigning „a maximal sequential path length in ***UUA***“ to ***UUA***, $\nu_{\#PORTS}(ip)$ is a mapping assigning „a number of all ***UUA*** ports that can be used for adjusting an *i path* in ***UUA***“ to ***UUA***.

Let $\iota_{CO} : \overline{\boldsymbol{IP}}_{\boldsymbol{UUA}} \rightarrow \mathcal{R}$ be a mapping assigning “an ease of adjusting an i path” to an i path defined

$$\iota_{CO}(ip) = \left(1 - \frac{\iota_{\#PORTS}(ip)}{\nu_{\#PORTS}+1}\right) \times \left(1 - \frac{\iota_{\#CLKS}(ip)}{\nu_{\#CLKS}+1}\right) \times \left(\prod_{\substack{(a,b)\in c,\\ c\in \iota_{IPAS}(ip)}} \pi_{CON}(a) \right). \quad (1)$$

Having mapping (1) evaluating i path numerically, we can easy select the best i path (i.e. i path with ι_{CO} closest to 1 value) between two ports and using this value, port controllability, observability and testability can be evaluated.

Let $\pi_{CON} : \boldsymbol{PORT_{UUA}} \rightarrow \mathcal{R}$ be a mapping assigning a controllability to a port p defined $\pi_{CON}(p) = \max_{\substack{pin \in \boldsymbol{PIN_{UUA}}, \\ ip \in \overline{IP}^{p}_{pin}}} (\iota_{CO}(ip))$. (2)

Let $\pi_{OBS} : \boldsymbol{PORT_{UUA}} \rightarrow \mathcal{R}$ be a mapping assigning an observability to a port p defined $\pi_{OBS}(p) = \max_{\substack{pout \in POUT_{UUA}, \\ ip \in \overline{IP}^{pout}_{p}}} (\iota_{CO}(ip))$. (3)

Let $\pi_{TST} : \boldsymbol{PORT_{UUA}} \rightarrow \mathcal{R}$ be a mapping assigning a testability to a port defined $\pi_{TST}(p) = \frac{\pi_{CON}(p) + \pi_{OBS}(p)}{2}$. (4)

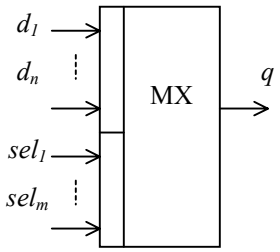
Relations (1)-(4) presented above are adequate for evaluating controllability, observability and testability values of each ***UUA*** port. In addition, relations (5)-(11) for evaluating output port controllability using input port controllability values and similarly relations for evaluating input port observability using output port observability value(s) are presented in this paper.

Before presenting these relations, let a mapping $\pi_{CO} : \boldsymbol{PORT_{UUA}} \times \mathcal{N} \times \mathcal{N} \rightarrow \mathcal{R}$ be defined $\pi_{CO}(p, m, n) =$

$$\left(1 - \frac{\iota_{\#PORTS}(ip) + m}{v_{\#PORTS} + 1}\right) \times \left(1 - \frac{\iota_{\#CLKS}(ip) + n}{v_{\#CLKS} + 1}\right) \times \left(\prod_{\substack{(a,b) \in c, \\ c \in \iota_{IPAS}(ip)}} \pi_{CON}(a)\right)$$

where $ip \in \overline{\boldsymbol{IP}}_{\boldsymbol{UUA}}$ and $\iota_{CO}(ip) = \pi_{CON}(p)$. (5)

There are three types of components in our digital circuit model at RT level – functional units, registers and multiplexers. Because these relations are component-type dependent, relations for each component type will follow.



Let the multiplexer has n input data ports $d_1, d_2, \ldots, d_n$, m input control ports $sel_1, sel_2, \ldots, sel_m$ and one output port q.

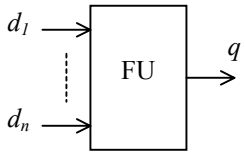
A formula (6) for evaluating a controllability of a multiplexer output port is presented. Then, $con(q)=0$ if a controllability of all d_i is equal to zero or if there is a sel_j that $con(sel_j)=0$. Otherwise $con(q) \in (0; 1)$.

$$\pi_{CON}(q) = 0 \text{ if } \forall d_i, i \in \{1,2,\ldots,n\} : \pi_{CON}(d_i) = 0 \\ = \max_{i=1}^{n} [\pi_{CO}(d_i, m, 0) \times \prod_{j=1}^{m} \pi_{CON}(sel_j)] \text{ otherwise} \quad (6)$$

A formula (7) for evaluating a controllability of a multiplexer output port is presented. If an observability of q is equal to zero or if sel_j exists that $con(sel_j)=0$ then $obs(d_i)=0$. Otherwise $obs(d_i) \in (0; 1)$.

$$\pi_{obs}(d_i) = 0 \text{ if } \pi_{obs}(q) = 0 \\ = \pi_{CO}(q, m, 0) \times \prod_{j=1}^{m} \pi_{CON}(sel_j) \text{ otherwise} \quad (7)$$

Let the FU has n input ports $d_1, d_2, \ldots, d_n$ and one output port q.



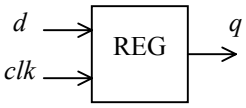
Then (8), $con(q)=0$ in the case there is no *i path* from any FU input port d_j to q or if a d_j exists that $con(d_j)=0$. Otherwise $con(q) \in (0; 1)$. The same formula can be also used for functional unit with more outputs.

$$\pi_{CON}(q) = 0 \text{ if } \forall d_i, i \in \{1, \ldots, n\} : (d_i, q) \notin \boldsymbol{I} \\ = \max_{i=1}^{n} [\pi_{CO}(d_i, n-1, 0) \times \prod_{\substack{j=1, \\ j \neq i}}^{n} \pi_{CON}(d_j)] \text{ otherwise} \quad (8)$$

A formula for (9) evaluating observability of one-output FU input port is presented. If there is no *i path* from d_i to q or if d_j exists that $con(d_j)=0$ or if $obs(q)=0$ then $obs(d_i)=0$. Otherwise $obs(d_i) \in (0; 1)$. This formula can be also generalized for functional unit with more outputs.

$$\pi_{OBS}(d_i) = 0 \text{ if } [\pi_{OBS}(q) = 0 \text{ or } (di, q) \notin \boldsymbol{I}] \\ = \pi_{CO}(q, n-1, 0) \times \prod_{\substack{j=1, \\ j \neq i}}^{n} \pi_{CON}(dj) \text{ otherwise} \quad (9)$$

Let *d* be a parallel data input port, *clk* be a clock input port and *q* a parallel data output port.



Then, a formula (10) for evaluating a controllability of a *q* is presented. It can be seen that *con(q)*=0 if a controllability of *d* or controllability of *clk* is equal to zero. Otherwise *con(q)* ∈ (0; 1).

$$\pi_{CON}(q) = 0 \text{ if } \pi_{CON}(d) = 0 \\ = \pi_{CO}(d,1,1) \times \pi_{CON}(clk) \quad \text{otherwise} \tag{10}$$

A formula (11) for evaluating an observability of a register input port *d* is presented. It can be seen that *obs(d)*=0 if an observability of *q* or controllability of *clk* is equal to zero. Otherwise *con(q)* ∈ (0; 1).

$$\pi_{obs}(d) = 0 \text{ if } \pi_{obs}(q) = 0 \\ = \pi_{CO}(q,1,1) \times \pi_{CON}(clk) \text{ otherwise} \tag{11}$$

Let a mapping (12) $\nu_{CON} : \varnothing \rightarrow \mathcal{R}$ assigning "average controllability of ***UUA*** nodes" to ***UUA*** be defined

$$\nu_{CON} = \frac{\sum_{p \in PORTS_{UUA}} \pi_{CON}(p)}{|PORTS_{UUA}|}. \tag{12}$$

Let a mapping (13) $\nu_{OBS} : \varnothing \rightarrow \mathcal{R}$ assigning "average observability of ***UUA*** nodes" to ***UUA*** be defined

$$\nu_{OBS} = \frac{\sum_{p \in PORTS_{UUA}} \pi_{OBS}(p)}{|PORTS_{UUA}|}. \tag{13}$$

Let a mapping (14) $\nu_{TST} : \varnothing \rightarrow \mathcal{R}$ assigning "average testability of ***UUA*** nodes" to ***UUA*** be defined

$$\nu_{TST} = \frac{\sum_{p \in PORTS_{UUA}} \pi_{TST}(p)}{|PORTS_{UUA}|}. \tag{14}$$

Having graph model presented in section 2 and formulas (1)-(4) or (5)-(11) for port (local) controllability, observability and testability evaluation, global controllability, observability and testability values can be assigned to ***UUA*** using formulas (12)-(14).

Experimental results gained using proposed formulas are presented in the next section.

4. Experimental Results

Testability analysis based on proposed testability measures (presented in the previous section) was verified on *DIFFEQ* benchmark circuit.

There are 71 ports in original *DIFFEQ* circuit structure – 12 ports are primary ports (11 inputs, 1 output) other 59 ports are ports of *DIFFEQ* inner elements. The "Port", "C", "O" and "T" symbol respectively in the heading of the following table means port name, port controllability value, port observability value and port testability value.

Port	C	O	T	Port	C	O	T
DIFFEQ.a	1	0	0.5	Mul1.a	.98	0	.493
DIFFEQ.dx	1	0	0.5	Mul1.b	.78	0	.394
DIFFEQ.clk	1	0	0.5	Mul1.q	.76	0	.383
DIFFEQ.const2	1	0	0.5	Reg4.d	.76	0	.383
DIFFEQ.sel1	1	0	0.5	Reg4.clk	1	0	0.5
DIFFEQ.sel2	1	0	0.5	Reg4.q	.64	0	.323
DIFFEQ.sel3	1	0	0.5	Sub.a	.64	0	.323
DIFFEQ.sel4	1	0	0.5	Sub.b	0	0	0
DIFFEQ.sel5	1	0	0.5	Sub.q	0	0	0
DIFFEQ.sel6	1	0	0.5	Reg1.d	0	0	0
DIFFEQ.sel7	1	0	0.5	Reg1.clk	1	0	0.5
DIFFEQ.result	0	1	0.5	Reg1.q	0	0	0
Mux1.a	.646	0	.32	Mux6.a	1	0	0.5
Mux1.b	1	0	0.5	Mux6.b	.79	0	.39
Mux1.sel	1	0	0.5	Mux6.sel	1	0	0.5
Mux1.q	.986	0	.49	Mux6.q	.98	0	.493
Mux3.a	0	0	0	Mux5.a	0	.798	.399
Mux3.b	1	0	0.5	Mux5.b	0	.798	.399
Mux3.sel	1	0	0.5	Mux5.sel	1	0	0.5
Mux3.q	.986	0	.49	Mux5.q	0	.81	.405
Mux4.a	1	0	0.5	Add.a	0	.81	.405
Mux4.b	0	.798	.39	Add.b	.98	0	.493
Mux4.c	0	.798	.39	Add.q	0	.833	.417
Mux4.sel1	1	0	0.5	Reg6.d	0	.833	.417
Mux4.sel2	1	0	0.5	Reg6.clk	1	0	0.5
Mux4.q	.972	0	.48	Reg6.q	0	.986	.493
Mul2.a	.986	0	.49	Comp.a	1	0	0.5
Mul2.b	.972	0	.48	Comp.b	0	.986	.493
Mul2.q	.945	0	.47	Comp.q	0	1	0.5
Reg5.d	.945	0	.47	Reg2.d	0	.986	.493
Reg5.clk	1	0	0.5	Reg2.clk	1	0	0.5
Reg5.q	.799	0	.39	Reg2.q	0	.798	.399
Mux2.a	0	0	0	Reg3.d	0	.986	.493
Mux2.b	.799	0	.39	Reg3.clk	1	0	0.5
Mux2.sel	1	0	0.5	Reg3.q	0	.798	.399
Mux2.q	.787	0	.39				

Table 1 – *DIFFEQ* port evaluation using proposed metrics

In the table 1, the controllability, observability and testability of all *DIFFEQ* ports are presented. It can be seen, that there

are 49 controllable ports, 16 observable ports and 65 testable ports in original *DIFFEQ* structure.

Global *DIFFEQ* parameters based on formulas presented in previous section are presented in the table 2. They show average ease of adjusting i path in *DIFFEQ* numerically. Since their values are much lower than 1, it can be stated that a possibility of adjusting i path in *DIFFEQ* is very difficult task.

Global Controllability	Global Observability	Global Testability
0.662	0.192	0.427

Table 2 – Global *DIFFEQ* parameters

5. Conclusion

Quite new 1) RTL digital circuit model for a DFT purposes and 2) testability measures defined over this model were presented in this paper.

Proposed testability metrics are used to evaluate testability level of each circuit node by a value from <0; 1> interval. This is needed for future development of a cost function evaluating a quality of a solution proposed by a DFT process and consequently also for a DFT process optimization.

Acknowledgements

This work has been financially supported by the Czech Ministry of Education–FRVŠ No. 1754/2002/G1–"Application of Evolution Approaches for Digital Circuit Testability Enhancement".

References

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

[2] Bukovjan, P.: Allocation for Testability in High-Level Synthesis, PhD thesis, Institut National Polytechnique de Grenoble, 2000

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

[4] Mao W., Ciletti M. D.: 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, June 12-15, 1988, Anaheim, CA, USA, pp. 591-596

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

[6] Kotásek Z., Růžička R., Strnadel J.: 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, HU, SU, 2001, pp. 123-128

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

[8] Abadir, M. S., Breuer, M. A.: A Knowledge Based System for Designing Testable VLSI chips, IEEE Design&Test, August 1985, pp. 56 – 68