Normalized Testability Measures at RT Level: Utilization and Reasons for Creation

Josef Strnadel*
strnadel@fit.vutbr.cz

Zdeněk Kotásek*
kotasek@fit.vutbr.cz

Abstract: The paper deals with a design overview of special-property factors for testability valuation of a digital circuit at RT level. The reasons for development of such factors and a way of their utilization are presented in this paper. Then, mathematical formulas are used to demonstrate these factors in a formal way and finally experimental results are presented.

Key Words: RT level, controllability, observability, testability, testability analysis

1 Introduction

The problem of digital circuit testing appeared a short time after first digital devices were created. Already in these times, there was a need to verify the structure/behavior correctness of a manufactured digital device comparing it with a structure and a simulated behavior of digital circuit developed by a designer. The basic effort of testing is to detect possible physical faults in a circuit structure causing wrong behavior of a manufactured circuit or of any of its parts. For a given fault model, there is a certain amount of faults in the digital circuit that can be (in case of real physical fault of this circuit) detected or also localized by testing the circuit. The percentage of physical faults (from total amount of faults in a given fault model) that can be detected/localized by a given test is called *fault coverage* and is one of the major test properties. The other important test property is a *test sequence* length, which means amount of *test vectors* needed for this test. A *test time* needed for applying test on given circuit is affected by this test property. It is requested to have a test with fault coverage as closer to 100% as possible achieved by an acceptable amount of test vectors. It is evident that it is needed to find a trade-off between the fault coverage and amount of test vectors. Generally, a test system consists 1) of a *test generator* (external, internal – LFSR, BILBO, HILDO etc.) generating a test sequence consisting of input test vectors and 2) of a *test responses analyzer* analyzing correctness of responses on input test vectors. A component consisting of a test generator and a test responses analyzer can be called a *tester*, which can be either external or internal. From a test design viewpoint, a proper test sequence is identified by a software tool first and then a test generator and a test responses analyzer are realized by a hardware. For simplification, we don't deal with more detail information about testing (test identification and generation types, test optimization techniques etc.) in this text.

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

* Faculty of Information Technology, Brno University of Technology, Božetěchova 2, CZ-612 66 Brno

test vectors, structures, methods etc.) can be designed manually and especially for this 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 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.

Usually, usage of DFT technique(s) is preceded by a process called *testability analysis*. For illustration of testability analysis goals, let us specialize on a *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 a testability level of this node to it. In our 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 [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 another topic.

2 Properties of Proposed Testability Measures

Before presentation of mathematical formulas for testability measures, basic demands on the measures will follow. Let us specialize on claims posed on testability values. These claims led to the current form of 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 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 is evaluated. Controllability, observability and testability values are supposed to be real numbers from <0; 1> interval.

First claim is posed on controllability values (see Fig. 1). 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 (see Fig. 1). 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.

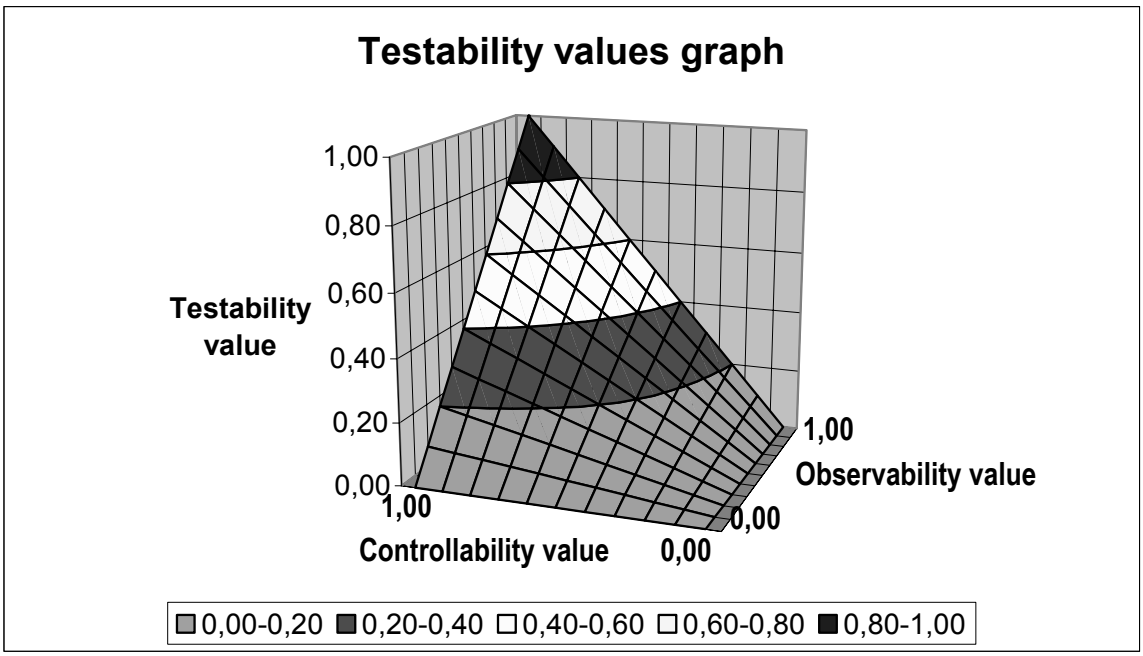


Fig. 1 – Testability values graph

Last claim is posed on testability values (see Fig. 1). 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.

3 Mathematical Formulas

After a presentation of basic requirements on controllability, observability and testability values, mathematical formulas fulfilling these claims can be presented. There are three types of components in our digital circuit model at RT level – functional units, registers and multiplexers. Proposed testability analysis is based on formulas for evaluating controllability of output nodes, formulas for evaluating an observability of input nodes and propagation process. The formulas are used for a forward controllability propagation and backward observability propagation respectively. For more detail information about this process see the next section.

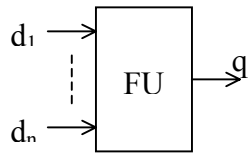
With a respect to a component type a node belongs to, there are 1) three formula types for evaluating a controllability of an component output node by a help of controllability of component input node(s) and 2) three formula types for evaluating an observability of a

component input node by a help of observability of component output node(s). Since a mathematical model of a digital circuit at RT level is not introduced in this paper due to a limited space of this paper, formulas will be presented with a help of a natural language for clearer understanding of principles they are built on.

Suppose that *NODES* set is a set of all digital circuit nodes and *R* is a set of real numbers. Let a function for evaluating a node controllability be defined as a mapping $con : NODES \to R$, a function for evaluating a node observability be defined as a mapping $obs : NODES \to R$ *and* a function for evaluating a node testability be defined as a mapping $tst : NODES \to R$.

3.1 Formulas for Controllability Evaluation

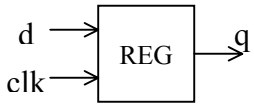
Let us start with three formulas for evaluating a controllability of output nodes. First, a formula for evaluating controllability of one-output functional unit (FU) output port is presented in Fig. 2. Let the FU has *n* input ports $d_1, d_2, \ldots, d_n$ and one output port *q*. Then, $con(q)=0$ in the case there is no *i path* [8] 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.



$$con(q) = 0 \quad \text{if no i path from any } d_i \text{ to } q \text{ exists, where } i = 1,2,\ldots n$$
$$= \max_{i=1}^{n} [\underset{+(n-1)\,nodes}{con}(d_i) \times \prod_{\substack{j=1,\\ j\neq i}}^{n} con(d_j)] \qquad \text{otherwise}$$

Fig. 2 – Formula for evaluating controllability of a FU output node

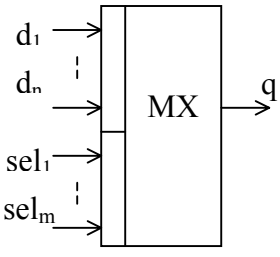
Let *d* be a parallel data input port, *clk* be a clock input port and *q* a parallel data output port. Then, a formula for evaluating a controllability of a *q* is presented in Fig. 3. It can be seen that $con(q)=0$ if a controllability of *d* or controllability of *clk* is equal to zero. Otherwise $con(q) \in (0; 1)$.



$$con(q) = 0 \quad \text{if } con(d) = 0$$
$$= \underset{\substack{+1\,node\\ +1\,clk}}{con}(d) \times con(clk) \qquad \text{otherwise}$$

Fig. 3 – Formula for evaluating controllability of a register output node

A formula for evaluating a controllability of a multiplexer output port is presented in Fig. 4. 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*. 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 $con(q) \in (0; 1)$.

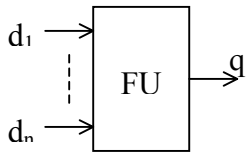


$$con(q) = 0 \quad \text{if } con(d_i) = 0 \text{ for any } i \in \{1,2,\ldots, n\}$$
$$= \max_{i=1}^{n} [\underset{+m\,nodes}{con}(d_i) \times \prod_{j=1}^{m} con(sel_j)] \qquad \text{otherwise}$$

Fig. 4 – Formula for evaluating controllability of a multiplexer output node

3.2 Formulas for Observability Evaluation

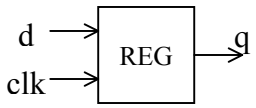
In the following text, three formulas for evaluating an observability of input nodes will be presented. First, a formula for evaluating observability of one-output FU input port is presented in Fig. 5. If there is no *i path* from d_i to q or if $d_j \neq d_j$ exists that $con(d_j)$=0 or if $obs(q)$=0 then $obs(d_i)$=0. Otherwise $obs(d_i) \in (0; 1)$. This formula can be also generalized for functional unit with more outputs.



$$obs(d_i) = 0 \quad \text{if } obs(q) = 0 \text{ or}$$
$$\text{if no i path from } d_j \text{ to } q \text{ exist}$$
$$= \underset{+(n-1)\ nodes}{obs}(q) \times \prod_{\substack{j=1,\\ j\neq i}}^{n} con(d_j) \qquad \text{otherwise}$$

Fig. 5 – Formula for evaluating observability of a FU input node

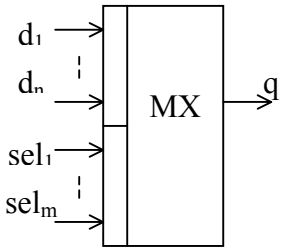
In Fig. 6, a formula for evaluating an observability of a register input port *d* is presented. It can be seen that $obs(d)$=0 if an observability of q or controllability of *clk* is equal to zero. Otherwise $con(q) \in (0; 1)$.



$$obs(d) = 0 \quad \text{if } obs(q) = 0$$
$$= \underset{\substack{+1\,clk\\ +1\,node}}{obs}(q) \times con(clk) \quad \text{otherwise}$$

Fig. 6 – Formula for evaluating observability of a register input node

Finally, a formula for evaluating a controllability of a multiplexer output port is presented in Fig. 7. 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 $obs(d_i) \in (0; 1)$.



$$obs(d_i) = 0 \quad \text{if } obs(q) = 0$$
$$= \underset{+m\ nodes}{obs}(q) \times \prod_{j=1}^{m} con(sel_j) \qquad \text{otherwise}$$

Fig. 7 – Formula for evaluating observability of a multiplexer input node

Because of limited space of this paper and an absence of digital circuit model, a graph model dependent sub-formulas used in previous six formulas will be presented only in brief and with a help of a natural language just for understanding of principles they are built on.

Generally, if we need to set up (control) a value on a given node *n,* we need to set a value on some primary input port first and then propagate this value through the circuit structure towards the node *n* by setting up values on other nodes and generating clock cycles. Similarly, if we need to observe a value on a given node *n*, we need to propagate this value through the circuit structure from node *n* towards circuit primary outputs by setting up values on other nodes and generating clock cycles. It can be seen that a difficulty of setting up or observing a

value on a node can be evaluated having information about a number of nodes that must be set up and a number of clock cycles that must be generated to control or observe a node value.

First, let us informally define some new terms. Let CC_n, respectively CO_n be values indicating a number of nodes that must be set up for controlling, respectively observing a value on a node n. Similarly, let SC_n, respectively SO_n be values indicating a number of clock cycles that must generated for controlling, respectively observing a node n value.

3.3 Transfer Sub-Formulas

In the Fig. 8, two simplified sub-formulas are be presented – the first one is used for evaluating a controllability transferred (with a certain loss/penalty representing overhead costs needed for data transfer through a component structure) to a component output port from component input ports and the second one for evaluating an observability transferred (with a certain loss) from a component output port to component input ports. Both these formulas are generalized for using with any component type at RT level.

$$\underset{\substack{+m\ nodes\\+n\ clks}}{con(d)} = \left(1 - \frac{CC_d + m}{nodes + 1}\right) \times \left(1 - \frac{SC_d + n}{lspl + 1}\right) \times ICP_d \qquad \underset{\substack{+m\ nodes\\+n\ clks}}{obs(q)} = \left(1 - \frac{CO_q + m}{nodes + 1}\right) \times \left(1 - \frac{SO_q + n}{lspl + 1}\right) \times ICP_q$$

a) b)

Fig. 8 – Sub-formulas for evaluating node a) controllability and b) observability

In Fig. 8, there are three terms that haven't been introduced yet. First one is a *nodes* term, which represents a number of non-primary input ports in given digital circuit. These are ports that are in conjunction with generating clock cycles used for controlling or observing other circuit nodes by adjusting an i path among tested node and some circuit primary input or output. Second one is a *lspl* term, which represents the longest sequential path length in given digital circuit. The last one are an ICP_d or ICP_q term respectively, which represent a product of controllabilities of all nodes that must be set up for controlling node d or observing node q respectively.

3.4 Formula for Testability Evaluation

Once both controllability (*con(n)*) and observability (*obs(n)*) value are assigned to each circuit node n, also testability value (*tst(n)*) can be can be assigned to it. Proposed testability value (see Fig. 9) of a node n is defined as an arithmetic average of its controllability and observability values.

$$tst(n) = \frac{con(n) + obs(n)}{2}$$

Fig. 9 – Formula for evaluating node testability

3.5 Formulas for Global Measures Evaluation

In a previous text, testability measures for evaluating digital circuit nodes were presented. These measures are also called *local measures*, because they are used for evaluation of node testability properties (respecting their controllability and observability properties). On the basis of these local measures *global measures* for evaluating testability properties of the entire circuit are constructed. Two types of global testability factors are differed in our approach. First one (see Fig. 10) includes global controllability, observability and testability respectively that are equal to arithmetic average of controllabilities, observabilities and testabilities of circuit nodes respectively.

$$con_{circuit} = \frac{\sum_{n \in NODES} con(n)}{|NODES|} \qquad obs_{circuit} = \frac{\sum_{n \in NODES} obs(n)}{|NODES|} \qquad tst_{circuit} = \frac{\sum_{n \in NODES} tst(n)}{|NODES|}$$

a)controllability b)observability c)testability

Fig. 10 – Formulas for evaluating global testability measures

Second one includes an information about an absolute or relative number of (un)controllable, (un)observable and (un)testable nodes from total number of all circuit nodes.

4 Testability Analysis Based on Proposed Formulas

Having all required formulas, a principle of a testability analysis on these formulas basis will be briefly presented in this section.

First, controllability, observability and testability value of each circuit node is initialized by setting to zero.

Then controllability of all primary inputs and of all nodes connected to them by a link is set to one. Alike, observability of all primary outputs and of all nodes connected to them by a link is set to one.

As a next stage, propagation of controllabilities in forward direction from primary inputs towards primary outputs is started. Using formulas presented in Fig. 2 – 4, controllability values are (with certain loss/penalty representing overhead costs needed for data transfer through a component structure) propagated from component inputs to its output(s). Then, these output controllabilities are propagated by links to input ports that haven't been processed yet and the process repeats. If there is no way of propagation, this stage will end and next one will start.

Alike, propagation of observability values in backward direction is done using formulas presented in Fig. 5 – 7.

As a last stage, testability of each circuit node is evaluated using formula presented in Fig. 9, then global testability factors are evaluated using formulas presented in Fig. 10 and numbers of uncontrollable, unobservable and untestable circuit nodes are identified.

5 Experimental Results

Testability analysis based on proposed testability measures was verified on *DIFFEQ* benchmark circuit. Total number of nodes in *DIFFEQ* is 71. Because of limited space, only global testability measures are presented. Results are summarized in Tab. 1.

Global Controllability	Global Observability	Global Testability	№ of uncontrollable nodes	№ of unobservable nodes	№ of untestable nodes
0.651	0.197	0.424	22	55	6

Tab. 1 – *DIFFEQ* testability results

6 Conclusions

A quite new testability measures were presented in this paper. They are used to evaluate a testability level of each digital circuit node by a normalized value from <0; 1> interval. This is needed for a future development of a normalized fitness function that will be used for evaluation of a solution proposed by a DFT process. Except the testability of digital circuit nodes also area and pin overheads, fault coverage and a test application time will be taken into account in this function. Alike a testability value of a node represents its testability level, a fitness value represents a quality level of a solution proposed by a DFT process.

Proposed testability metrics are used for evaluation of testability qualities of all circuit nodes. Consequently, testability level of a given digital circuit is evaluated. For more accurate testability evaluation of a circuit, also testability evaluation of links could be needed. Also this fact will be one of future topics we will deal with.

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" and the Research Intent of FEI, Brno University of Technology, Czech Republic, grant No. CEZ: J22/98:262200012 "Research of the Information and Control Systems".

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 en vue de la testabilité dans le cadre de la synthèse de haut niveau, 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, ISBN 80-85988-57-7
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, ISBN 963-7175-16-4
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