Computer Science and Information Systems 15(1):237–256 https://doi.org/10.2298/CSIS161118040B

Black Box Delay Fault Models for Non-scan Sequential Circuits

Eduardas Bareisa[1], Vacius Jusas[1], Kestutis Motiejunas[1], Liudas Motiejunas[2], and Rimantas Seinauskas[1],

[1] Kaunas University of Technology, Studentu 50-404,
LT-51368 Kaunas, Lithuania
{eduardas.bareisa, vacius.jusas, kestutis.motiejunas, rimantas.seinauskas}@ktu.lt
[2] Kaunas University of Technology, Studentu 50-401,
LT-51368 Kaunas, Lithuania
liudas.motiejunas@ktu.lt

Abstract. We presented nine new black box delay fault models for non-scan sequential circuits at the functional level, when the primary inputs and primary outputs are available only. We examined the suggested fault models in two stages. During the first stage of the experiment, we selected the best two fault models for further examination on the base of criterion proposed in the paper. During the second stage, we used the functional delay fault model and two black box delay fault models from the first stage for test selection. The comparison of fault coverages was carried out for transition faults. The obtained results demonstrate that transition fault coverages of tests selected based on proposed black box fault models are similar to coverages of tests selected based on functional delay fault model that uses the inner state of circuit.

Keywords: sequential non-scan circuit, functional test, black box delay fault model.

1. Introduction

Test generation problem for sequential circuits is intractable. Therefore, the scan register is additionally introduced into the circuits, which significantly facilitates the testing challenges. However, the use of scan register is associated with an increase in circuit scale and duration of the testing process [1]. Therefore, the test generation for some non-scan circuits still remains relevant.

Modern automatic test generation tools can generate tests only for small-scale circuits. Despite considerable efforts testing community still unable to develop effective methods to find the right solutions in the large search area. Deterministic test generation methods usually cannot find the solution, when the fault detection requires long test sequence. Therefore, the functional test generation methods based on circuit model at a high level of abstraction are used. The random test generation methods are based on the test sequence template. The template determines the length of the test sequence and the use of the reset signal. Therefore, the test sequence length for the random generation method is not a critical parameter. Evolutionary generation methods are based on evolutionary mutations on already available test sequences. Additionally, genetic algorithms use the crossing operations. In all cases, evaluation of the eligibility of the test sequence is essential.

Fault simulation tools quite accurately indicate the faults detectable by the test sequence. However, fault simulation tools require significant computing resources and little amount of test sequences can be examined during test generation based on fault simulation. Therefore, the fault models at a high level of abstraction are needed. Functional fault model allows to examine more test sequences during the same period of time, but the usefulness of the test sequences has to be clarified with fault simulation at a gate level after test generation.

Functional delay fault models based on the primary inputs, primary outputs, and state variables of the circuit have been successfully used for test generation [2], [3], [4], [5], [6], [7]. Functional delay fault model is based on the stuck-at fault pairs at the primary inputs, primary outputs and state variables. Circuit model, which represents single clock cycle operations, is used. Device models at the higher level of abstraction, which do not have clock information, are used in the initial design stages. In this case, information about the state variables is not available yet and fault models can rely on the primary inputs and primary outputs of the circuit only. Such fault models are called black box fault models. Circuit models at the higher level of abstraction are more compact and, therefore, their use for test generation allows to obtain greater performance. As a result, more test sequences can be examined during the same time. Creation of the model on a higher level of abstraction requires less effort by a designer, as well. Localization of state variables requires considerable efforts, but there is no need to perform it when creating a black box model for delay faults.

This article aims to develop and explore black box delay fault models based solely on the primary inputs and primary outputs of circuit and to compare them with functional fault model that obtained quite satisfactory results of functional delay test generation [6]. The goal of the presented black box fault models is to replace the functional fault model. The obtained test sequences according to the proposed fault models can be used as a replacement for functional test sequences.

The rest of the paper is organized as follows. We review the related work in Section 2. We present several new functional black box delay fault models in Section 3. We introduce criterion in order to compare presented black box models in Section 4. We report the results of the investigation of the presented black box models for several circuits and we choose the best two models for further experiments in Section 5. We describe the results of the final experiment in Section 6. We finish with conclusions in Section 7.

2. Related Work

Delay test generation for sequential circuits can be broadly classified into two categories: structural level and functional level. Many approaches and tools are developed to construct delay test at the structural level. The best known approach is scan based testing [8], [9], [10]. Despite the high delay fault coverage, the structural testing suffers from several disadvantages. The overtesting is possible since scan architecture allows application of test sequences that are not possible during normal operation. The scan functioning mode requires long test application times. Moreover, the test generation is possible at the later stages of the design only when the structure of the circuit is synthesized already. To overcome these drawbacks the methods of functional delay test generation are developed.

Functional delay test sequences can be produced using functional delay fault models described in [2], [3], [4], [5], [6], [7]. In this case, test generation usually targets a specific fault model to ensure that the sequences are effective to detect delay faults at the gate level of the circuit. Alternatively, it is possible to use test sequences that were generated as part of a simulation-based design verification process. In this case, either various metrics [11], [12], [13], [14] are suggested how to select the appropriate test sequences from the large verification test pool or selection procedures [16], [17] are proposed in order to improve the effectiveness of the existing functional test sequences.

Kang et al. [2] suggested a register-transfer level metric for functional test selection from existing pool of functional test sequences. The metric is based upon an input/output transition (TRIO) fault model. The model is defined according to the primary inputs, primary outputs, and state variables of the module and it is based on a fault-free simulation only. During logic simulation the traces on the bits of interest are captured and then post-processed to get the TRIO fault coverage. But the model is approximate due to the following reasons: 1) it does not stipulate toggle propagation all the way to the primary outputs because it requires expensive fault simulation; 2) the evaluation of the transition at the output, which depends on multiple input transitions, is too much optimistic. The presented experimental results demonstrate quite a large loss of transition fault coverage of the initial test pool for some circuits.

Bareiša et al. [3] introduced three different new functional delay fault models. According to the proposed models, the functional delay faults are considered on the primary inputs, primary outputs and state bits of the model. The experimental results were presented for two small circuits only. Next, Bareiša et al. [4] presented an approach of test generation for non-scan synchronous sequential circuits using functional delay fault models. The software prototype model was used for definition of the function of the circuit. The non-scan sequential circuit was represented as the iterative logic array model consisting of k copies of the combinational logic of the circuit. The value k defined the number of clock cycles. But the presented approach of implementation of functional delay fault models was not efficient for large and complex circuits; it required long computation times.

The idea of functional delay fault models [4] was explored further and developed in [5]. The goal of the research was the efficiency of the implementation of the models, which was expressed in the obtained delay fault coverage and required computation time. The authors considered two functional fault models. The first model was restricted concerning the faults on the state bits; the pairs of stuck-at faults on the previous state bits and the primary outputs were considered only. The second fault model encompassed the first model, and additionally dealt with the pairs of stuck-at faults on the primary inputs and the next state bits, as well as, with the pairs of stuck-at faults on the previous state bits and the next state bits. The authors proposed two different implementations for these fault models: forward propagation and backward propagation. The experiments did not reveal the single best model, but the implementation of backward propagation required much less time for both models and obtained the same fault coverage in many cases as the implementation of forward propagation.

In order to improve the performance of test generation Bareisa et al. [6] suggested a method for functional delay test generation. The method consists of two following stages: preliminary test selection and functional test generation based on fault simulation [5]. The goal of the first stage is to minimize the number of test sequences considered during the second stage. The simplified functional delay fault model when the state bits are considered as the primary outputs is used during the first stage.

Sauer et al. [7] presented automated test pattern generation (ATPG) system for small-delay faults in non-scan circuits. The ATPG tool identifies the longest paths suitable for functional fault propagation and generates the shortest possible test sequences per fault. The flip-flops are considered during construction of the test sequence. The conventional scheme of the time frame is used. All the test sequences start at the circuit's initial state. The experimental results on the circuits from the ISCAS'89 and ITC'99 benchmark suites demonstrated the applicability of the proposed ATPG.

Pomeranz et al. [11] described a stuck-at fault coverage metric based only on logic simulation of the gate level circuit. The metric is based on the set of states that the circuit traverses under the test sequence. The authors [11] defined several versions of the metric suitable for different applications. The experimental results demonstrated the effectiveness of the metric for the ranking of test sequences based on their fault coverage.

Fang et al. [12] proposed and developed in [13] output deviations as a metric to grade functional test sequences at the register transfer level without explicit fault simulation. Experimental results for the open-source Biquad filter core and the Scheduler module of the Illinois Verilog Model showed that the deviations metric is computationally efficient and it correlates well with gate-level coverage for stuck-at, transition-delay, and bridging faults.

Vinutha et al. [14] described a metric to grade the test sequence using instruction-execution graph. The metric is based on the set of registers the circuit traverses under the test sequence. Using this information in combination with the observability and controllability of the register, the test sequence is graded. Experimental results on Parwan processor showed the effectiveness of the metric in ranking the test sequence based on their fault coverage.

Gent and Hsiao [15] proposed a control path aware, rule-based statement coverage metric at the Register Transfer Level to capture faulty behavior of statements along distinct operation paths within the circuit description. Experiments showed that the proposed metric has a strong correlation with gate level faults across a variety of benchmarks, including the microprocessor or 1200 with a power management unit. Additionally, the metric showed high level of scalability, providing up to two orders of magnitude reduction in execution time compared to fault simulation and up to an order of magnitude improvement over logic simulation based fault grading techniques.

Pomeranz [16] suggested a procedure that uses functional test sequences from verification test pool as a basis for forming a single functional test sequence. The procedure uses subsequences extracted from the end of the test sequences in the pool to take advantage of the ability of these subsequences to detect target faults when they are concatenated to the test sequence being formed. Fault simulation of single stuck-at faults is carried out on the target sequence only. Experimental results for transition faults demonstrated that the procedure produces a test sequence that detects at least as many faults as the pool to which it is applied. The length of the test sequence is in most cases significantly lower than the total length of the test sequences in the pool.

Pomeranz [17] proposed a procedure that reduces the functional test sequences by using pairs of sequences to produce additional sequences, referred to as overlaps. An overlap consists of the first vectors of one sequence and the last vectors of another. Overlaps are efficient in detecting target faults since they combine initialization, fault activation and fault propagation conditions from two sequences in different ways. During the construction of the test sequence, the target faults are single stuck-at faults, but the transition fault coverage is provided for the final test sequence, as well. The use of overlaps allows removing of some sequences from the initial test set. Consequently, the obtained size of the final test set and the overlaps are reduced.

Pomeranz [18] introduced a generalized definition of unnecessary test vectors, which allows additional ones to be omitted. According to this definition, a test vector is unnecessary if every target fault can be detected by a sequence that is obtained after omitting the vector, and possibly other vectors. The author developed a procedure for omitting test vectors based on the definition and discussed its effects on the storage requirements and test application time.

All the presented approaches so far use internal state directly in order to construct functional delay test for non-scan sequential circuits. We are suggesting the delay fault models for non-scan sequential circuits, which are based exceptionally on the primary inputs and primary outputs only without direct use of the internal state of the non-scan sequential circuit. To our best knowledge, there are no present similar approaches for testing hardware, but there exist some similar approaches for testing software [19], [20], [21].

Kant et al. [19] presented an approach based on genetic algorithms. According to the approach, the functional test cases are identified from functional requirements of the tested system that is considered as a mathematical function mapping its inputs onto its outputs. The purpose of the approach is to eliminate “bad” test cases that are unable to expose any error. The application of the approach is demonstrated on testing a complex Boolean expression. An approach based on genetic algorithms is described in [20], as well. Vos et al. [20] presented evolutionary testing framework to facilitate the development of evolutionary functional tests. Two case studies from industry are described. The biggest problem is the creation of the fitness functions, because it is a time-consuming. This problem prevents functional evolutionary testing from widespread in industry.

Fraser and Walkinshaw [21] offer an approach to incorporate behavioural coverage. The BESTEST approach enables using of the machine learning algorithms to augment standard syntactic testing approaches and applies search-based testing techniques to generate test sets with respect to the behavioural criterion. An empirical study on Java units demonstrates that test sets with higher behavioural coverage significantly outperform current baseline test criteria in terms of detected faults.

3. Black Box Delay Fault Models

We consider the non-scan synchronous sequential circuit. Let a circuit have a set of primary inputs $X = \{x_1, ..., x_i, ..., x_n\}$, and a set of primary outputs $Y = \{y_1, ..., y_j, ..., y_m\}$. In order to test such a circuit, test sequences of a defined length have to be applied. The length of test sequences can vary, but we will consider the test sequences of the same length for the same circuit for the simplicity of implementation. We denote by k the length of the test sequence.

The test sequence has a value for the detection of delay faults, if a transition at the primary input invokes a transition at the primary output. This idea is a base for the construction of the black box delay fault models. Therefore, our defined models always include a pair consisting of the primary input and the primary output, and values related to them. The models are defined as 4-tuples. We consider different alternatives to define the relationship between the transitions at the primary inputs and the primary outputs. The alternatives are as follows:

1. To define the relationship between separate patterns in the test sequence.
2. To define the relationship without considering the particular patterns in the test sequence.
3. To analyse either the transition of the value or change of the value at the primary output.
4. To count the number of the test patterns when the change of the value appears at the primary output after start of the transition at the primary input.
5. To use either fault-free transition simulation or transition fault simulation in the process of test sequence evaluation.
6. To define the pairs of the primary outputs having transitions without paying attention to the primary inputs.

In such a way, we develop nine new black box delay fault models. For the sake of the simplicity, we name them as M1-M9. The distinguishing feature of the models is a dependency of the number of faults on the length of test sequence. All the fault models are defined as 4-tuples $< x_{\mathrm{i}}, t_{\mathrm{i}}^{ii}, y_{\mathrm{j}}, t_{\mathrm{j}}^{jj}>$, where x_{i} is a primary input, t_{i}^{ii} is a rising (r_{i}^{ii}) or falling (f_{i}^{ii}) transition at x_{i} on clock period *ii* (2, …, k), y_{j} is a primary output, and t_{j}^{jj} is a rising (r_{j}^{jj}) or falling (f_{j}^{jj}) transition at y_{j} on clock period *jj* (2, …, k).

At the highest level of the abstraction (see Fig. 1), we can divide the fault models into two groups: the models based on fault-free transition simulation (M1, M2, M9) and the models based of transition fault simulation (M3-M8). According to the model M1, the pairs of primary inputs and primary outputs having transitions are considered taking into the number of clock period when transition happens. The model M2 is the same as the model M1; however, the number of clock period is not taken into account. Therefore, the model M2 is a compressed version of model M1. The model M9 is more different from the other models, because the values at the primary inputs are not considered; it is solely based on the values at the primary outputs.

According to the model M3, the transition at the primary input is changed to stable value, and the transition simulation is carried out. The changes of the transitions at the primary outputs, which occur due to this change, are considered. The model M4 is similar to the model M3, but all the changes including stable values are considered. The model M5 is a compressed version of the model M3 (the clock period is not taken into account).

The model M6 is a compressed version of the model M4 (the clock period is not taken into account). The model M7 is similar to the model M3; however, additionally, the number of the test patterns when the change of the value appears at the primary output is counted. The model M8 is a compressed version of the model M7 (the clock period is not taken into account).

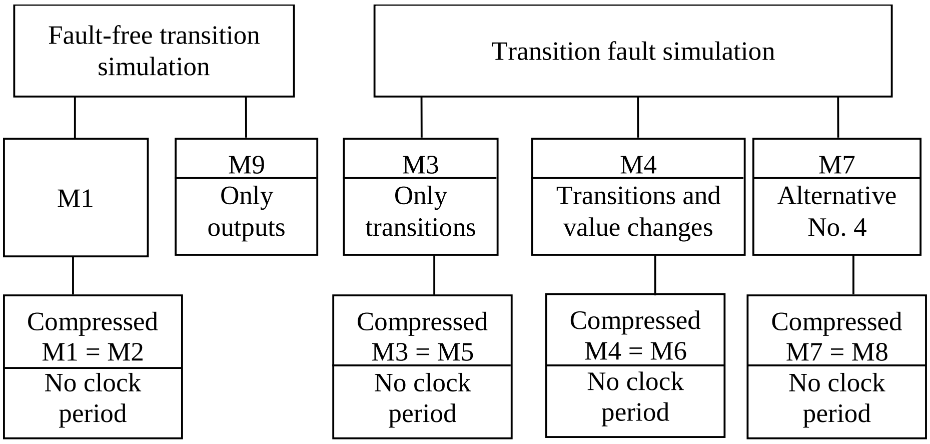

Fig. 1. Dependencies of black box delay fault models

Next, we provide the formal definitions of the proposed fault models.

Definition 1. M1 fault is a 4-tuple < x_i, t_i^{ii}, y_j, t_j^{jj} >, where x_i (i = 1, …, n) is a primary input, t_i^{ii} is a rising or falling transition at x_i on clock period ii (ii = 2, …, k), y_j (j = 1, …, m) is a primary output, and t_j^{jj} is a rising or falling transition at y_j on clock period jj >= ii (jj = 2, …, k). The M1 fault < x_i, t_i^{ii}, y_j, t_j^{jj} > is detected if transitions t_i^{ii} and t_j^{jj} are present in the logic simulation trace.

The model M1 is based on the assumption that the transition at the primary input x_i on the clock period ii invokes the transitions at the primary outputs, which occur on either the same clock period or the later clock periods. Such a condition is guaranteed by the constraint (jj >= ii). The maximum number of possible M1 faults is 4*n*m*(k*(k-1)/2). The members of this expression can be justified in the following way. The factor "4" is used since we have either rising or falling transition at the primary input and either rising or falling transition at the primary output (4 combinations); n – number of primary inputs; m – number of primary outputs; the subexpression (k*(k-1)/2) assesses the count of different combinations of primary input and primary output along the whole test sequence of length k.

Definition 2. M2 fault is a 4-tuple < x_i, t_i^{ii}, y_j, t_j^{jj} >, where x_i (i = 1, …, n) is a primary input, t_i is a rising or falling transition at x_i on any clock period ii <= jj (ii = 2, …, k), y_j (j = 1, …, m) is a primary output, and t_j^{jj} is a rising or falling transition at y_j on clock period jj (jj = 2, …, k). The M2 fault < x_i, t_i^{ii}, y_j, t_j^{jj} > is detected if transitions t_i^{ii} and t_j^{jj} are present in the logic simulation trace, and then the fault is represented as < x_i, t_i, y_j, t_j^{jj} >.

The maximum number of possible M2 faults is 4*n*m*(k-1). Let us consider an example. The results of fault-free simulation for hypothetical circuit having four primary inputs and three primary outputs are provided in 0. According to the fault model M1, the following faults are detected: $<x_1, r_1^2, y_1, r_1^2>$, $<x_1, r_1^2, y_3, f_3^4>$, $<x_1, f_1^3, y_3, f_3^4>$, $<x_1, r_1^4, y_3, f_3^4>$, $<x_2, r_2^3, y_3, f_3^4>$, $<x_2, f_2^4, y_3, f_3^4>$, $<x_4, f_4^2, y_1, r_1^2>$, $<x_4, f_4^2, y_3, f_3^4>$. According to the fault model M2, the following faults are detected: $<x_1, r_1, y_1, r_1^2>$, $<x_1, r_1, y_3, f_3^4>$, $<x_1, f_1, y_3, f_3^4>$, $<x_2, r_2, y_3, f_3^4>$, $<x_2, f_2, y_3, f_3^4>$, $<x_4, f_4, y_1, r_1^2>$, $<x_4, f_4, y_3,$

f_3^4>. If we compare these two lists, we notice that two faults <x_1, r_1^2, y_3, f_3^4>, <x_1, r_1^4, y_3, f_3^4> from fault model M1 are covered by single fault <x_1, r_1, y_3, f_3^4> in fault model M2, all the other faults from model M1 have correspondence one-to-one to the faults from model M2. For example, the fault <x_1, r_1^2, y_1, r_1^2> from model M1 corresponds to the fault <x_1, r_1, y_1, r_1^2> from model M2, and so on.

Table 1. Fault-free simulation

x_1	x_2	x_3	x_4	y_1	y_2	y_3
0	0	0	1	0	1	1
1	0	0	0	1	1	1
0	1	0	0	1	1	1
1	0	0	0	1	1	0

We already know that the list of the faults according to fault model M2 is a subset of the faults according to fault model M1. One could wonder why we are interested in the fault model that is a subset on the other fault model. The reasoning is the following. We have noticed many times that the test generation according to the faults at high level of design obtains the large test sets that are really redundant. Therefore, we suggest the compressed version of the fault model that would ensure the smaller test set. We will apply this practice for the other fault models, as well.

Definition 3. M3 fault is a 4-tuple < x_i, t_i^{ii}, y_j, t_j^{jj} >, where x_i (i = 1, …, n) is a primary input, t_i^{ii} is a rising or falling transition at x_i on clock period ii (ii = 2, …, k), y_j (j = 1, …, m) is a primary output, and t_j^{jj} is a rising or falling transition at y_j on clock period jj >= ii (jj = 2, …, k). The M3 fault < x_i, t_i^{ii}, y_j, t_j^{jj} > is detected if the elimination of the transition t_i^{ii} causes the disappearance of the transition t_j^{jj}.

By the term "elimination of the transition", we denote the application of the value at the primary input in the previous clock period to the current clock period in the case when these values are different. After the application of the value, the newborn test sequence is simulated. This process is a fault simulation. After simulation, some changes can be observed at the primary outputs. By the term "disappearance of the transition", we denote the situation when the transition of the signal was present on clock period jj at the primary output y_j before simulation, and this transition is absent after simulation of the applied value at the primary input. For example, consider the results of fault-free simulation in 0. For the primary input x_1 on the second clock period, we would assume the value 0 and we would simulate this changed test sequence. After simulation, we would compare the obtained results with the results in 0. For example, we would obtain the value 0 on the second clock period at the primary output y_1. Then the fault < x_1, t_1^2, y_1, t_j^2 > would be labeled as detected.

The faults in the model M3 are represented in the same way as in the model M1, but their detection is based already on fault simulation.

Definition 4. M4 fault is a 4-tuple < x_i, t_i^{ii}, y_j, c_j^{jj} >, where x_i (i = 1, …, n) is a primary input, t_i^{ii} is a rising or falling transition at x_i on clock period ii (ii = 2, …, k), y_j (j = 1, …, m) is a primary output, and c_j^{jj} is logic value 0 or 1 at y_j on clock period jj >= ii (jj = 2, …, k). The M4 fault < x_i, t_i^{ii}, y_j, c_j^{jj} > is detected if the elimination of the transition t_i^{ii} causes the change of the logic value c_j^{jj}.

The model M4 is similar to the model M3, but the detection condition is weakened. Instead of the condition that the elimination of the transition at the primary input would cause the disappearance of the transition at the primary output, it is enough that the elimination of the transition at the primary input would cause the change of the value at the primary output. In such a case, we obtain more faults; consequently, we will have more test sequences.

Definition 5. M5 fault is a 4-tuple $< x_\mathrm{i}, t_\mathrm{i}^{ii}, y_\mathrm{j}, t_\mathrm{j}^{jj} >$, where x_i ($i = 1, \ldots, n$) is a primary input, t_i is a rising or falling transition at x_i on any clock period $ii <= jj$ ($ii = 2, \ldots, k$), y_j ($j = 1, \ldots, m$) is a primary output, and t_j^{jj} is a rising or falling transition at y_j on clock period jj ($jj = 2, \ldots, k$). The M5 fault $< x_\mathrm{i}, t_\mathrm{i}^{ii}, y_\mathrm{j}, t_\mathrm{j}^{jj} >$ is detected if the elimination of the transition t_i^{ii} causes the disappearance of the transition t_j^{jj}, and then the fault is represented as $< x_\mathrm{i}, t_\mathrm{i}, y_\mathrm{j}, t_\mathrm{j}^{jj} >$.

Definition 6. M6 fault is a 4-tuple $< x_\mathrm{i}, t_\mathrm{i}^{ii}, y_\mathrm{j}, c_\mathrm{j}^{jj} >$, where x_i ($i = 1, \ldots, n$) is a primary input, t_i is a rising or falling transition at x_i on any clock period $ii <= jj$ ($ii = 2, \ldots, k$), y_j ($j = 1, \ldots, m$) is a primary output, and c_j^{jj} is a is logic value 0 or 1 at y_j on clock period jj ($jj = 2, \ldots, k$). The M6 fault $< x_\mathrm{i}, t_\mathrm{i}^{ii}, y_\mathrm{j}, c_\mathrm{j}^{jj} >$ is detected if the elimination of the transition t_i^{ii} causes the change of the logic value c_j^{jj}, and then the fault is represented as $< x_\mathrm{i}, t_\mathrm{i}, y_\mathrm{j}, c_\mathrm{j}^{jj} >$.

Definition 7. M7 fault is a 4-tuple $< x_\mathrm{i}, t_\mathrm{i}^{jj}, y_\mathrm{j}, t_\mathrm{j}^{jj+p} >$, where x_i ($i = 1, \ldots, n$) is a primary input, t_i^{jj} is a rising or falling transition at x_i on any clock period jj ($jj = 2, \ldots, k$), y_j ($j = 1, \ldots, m$) is a primary output, and t_j^{jj+p} is a rising or falling transition at y_j after p clock periods ($0 <= p <= k$-2 and $jj + p <= k$). The M7 fault $< x_\mathrm{i}, t_\mathrm{i}^{jj}, y_\mathrm{j}, t_\mathrm{j}^{jj+p} >$ is detected if the elimination of the transition t_i^{jj} causes the disappearance of the transition t_j^{jj+p}, and then the fault is represented as $< x_\mathrm{i}, t_\mathrm{i}, y_\mathrm{j}, t_\mathrm{j}^{p} >$.

The model M7 is modification of models M3 and M5, where the value p plays the key role. We will explain the motivation for this model using example. Let us have the following M3 faults: $< x_2, r_2^8, y_{15}, r_{15}^{13}>$, $< x_2, r_2^8, y_{15}, r_{15}^{18}>$, $< x_2, r_2^{24}, y_{15}, r_{15}^{29}>$. According to the model M7, these faults are labeled as follows: $< x_2, r_2^8, y_{15}, r_{15}^{8+5}>$, $< x_2, r_2^8, y_{15}, r_{15}^{8+10}>$, $< x_2, r_2^{24}, y_{15}, r_{15}^{24+5}>$. These three faults are represented by two following faults: $< x_2, r_2, y_{15}, r_{15}^{5}>$, $< x_2, r_2, y_{15}, r_{15}^{10}>$ in the model M5. The goal of the model M7 is to increase the diversity of the paths that would be covered in the circuit, since the model requires that the transition from the primary input would be observed at the primary output after different number of clock periods.

Definition 8. M8 fault is a 4-tuple $< x_\mathrm{i}, t_\mathrm{i}^{jj}, y_\mathrm{j}, c_\mathrm{j}^{jj+p} >$, where x_i ($i = 1, \ldots, n$) is a primary input, t_i^{jj} is a rising or falling transition at x_i on any clock period jj ($jj = 2, \ldots, k$), y_j ($j = 1, \ldots, m$) is a primary output, and c_j^{jj+p} is a logic value 0 or 1 at y_j after p clock periods ($0 <= p <= k$-2 and $jj + p <= k$). The M8 fault $< x_\mathrm{i}, t_\mathrm{i}^{jj}, y_\mathrm{j}, c_\mathrm{j}^{jj+p} >$ is detected if the elimination of the transition t_i^{jj} causes the change of the logic value c_j^{jj+p}, and then the fault is represented as $< x_\mathrm{i}, t_\mathrm{i}, y_\mathrm{j}, c_\mathrm{j}^{p} >$.

The model M8 is a modification of models M4 and M6. The maximum number of possible M5-M8 faults is the same as of M2 faults.

Definition 9. M9 fault is a 4-tuple $< y_\mathrm{i}, t_\mathrm{i}^{ii}, y_\mathrm{j}, t_\mathrm{j}^{jj} >$, where y_i ($i = 1, \ldots, m$) and y_j ($j = 1, \ldots, m$) are the primary outputs, t_i^{ii} is a rising or falling transition at y_i on clock period ii

(ii = 2, …, k), and t_{j}^{jj} is a rising or falling transition at y_{j} on clock period jj (jj = 2, …, k). The M9 fault < y_{i}, t_{i}^{ii}, y_{j}, t_{j}^{jj} > is detected if transitions t_{i}^{ii} and t_{j}^{jj} are present in the logic simulation trace.

The models M1-M8 were constructed on the base “cause-consequence” whereas the reason of the model M9 is to cover as many as possible combinations of values at the primary outputs. The maximum number of possible M9 faults is $4*m*m*k*k$.

4. Criterion to Compare Black Box Delay Fault Models

We will use the presented fault models to select test sequences from randomly generated ones. Therefore, we are going to define the criterion to compare the obtained results. Usually, three factors define the quality of test selection. These factors are as follows: the fault coverage, the selection time (speed) and the number of selected test patterns (in our case, the number of selected test sequences). The fault coverage is the most important one, but the other factors have to be considered as well. For example, we obtain compact test sequences during 5 minutes of test generation and we obtain large test sequences after 24 hours of test generation, and these test sequences have little higher fault coverage than the compact ones. We have to decide which obtained test generation result is better. Our goal is to combine the different factors of test quality evaluation (fault coverage, test selection time and number of selected test patterns) into single formula. Consequently, we provide an analytical expression that would help us to assess and to compare various fault models using one integrated number.

The proposed fault models represent lists of different faults. Therefore, the obtained fault coverage cannot be compared directly. Some common denominator has to be found. As the common denominator, we consider the obtained fault coverage of test sequences generated according to the fault model in question for some common type of delay faults. This is the first basic assumption in creating the analytical expression. The second basic assumption is formulated as follows. The fault model has some value if the amount of selected test sequences has the higher fault coverage than the same amount of randomly generated test sequences. This is the most important and robust factor. These two assumptions are the main base in the construction of analytical expression to compare different fault models.

Let us say, we have two fault models X and Y, and we wish to find out how well the test sequences generated according to fault model X detect faults of type Y. We denote randomly generated test set S that consists of n test sequences. The number of selected test sequences from set S according to fault model Y is m_Y ($m_Y <= n$) and these test sequences detect k_{YY} faults of type Y. We denote the test set S_X selected from S according to fault model X and the number of test sequences is m_X ($m_X <= n$). The fault simulation of test set S_X according to fault model Y shows that the number of test sequences, which detect faults Y, is m_{XY} ($m_{XY} <= m_X$), and the number of detected faults is k_{XY} ($k_{XY} <= k_{YY}$). The first m_X test sequences from randomly generated test set S detect the number k_{RY} ($k_{RY} <= k_{YY}$) of faults Y, and T_X is the time of test generation using fault model X; the time is measured in seconds.

Next, we denote the fault coverage of the selected test sequences

$$C_X = \frac{k_{XY}}{k_{YY}} * 100, \tag{1}$$

and the fault coverage of randomly generated test sequences

$$C_R = \frac{k_{RY}}{k_{YY}} * 100. \tag{2}$$

We assume that the main issue in the assessing the quality of fault model is its ability to select test sequences meaningfully and, therefore, we compare the coverage of the selected test set with the coverage of randomly generated test set in the nominator (formula (3)). All the other factors can decrease this quality only. They can be incorporated into the analytical expression in different ways. We have chosen to use the division. We divide the difference of fault coverage $C_X - C_R$ by ratio of number of test sequences and the logarithm of time. The ratio $\frac{m_X}{m_{XY}}$ shows the redundancy of selected test sequences. We suppose that the factor of time is not so important. Therefore, we apply the logarithm that allows decreasing the impact of time. However, that is not enough. It is important to know, as well, how the selected test sequences are able to detect the faults of type *Y*. The expression $(100 - C_X)$ assess the ability to fully cover the faults of type *Y*. The lesser the value is, the better test sequence is. We add 1 to this expression that the value would not become equal to 0 in case of full coverage. Such situation is almost impossible, but for the safety sake.

Using introduced assumptions and conventions, we suggest the following formula to compare different fault models:

$$Q_X = \frac{\frac{C_X - C_R}{\frac{m_X}{m_{XY}} + \ln{(T_X)}}}{(100 - C_X) + 1} \quad . \tag{3}$$

We provide some short comments for the formula. The expression $C_X - C_R$ allows comparing the obtained fault coverage with the fault coverage of the same amount of randomly generated test sequences. The expression $\frac{m_X}{m_{XY}}$ shows the redundancy of the generation. The expression $\ln{(T_X)}$ assess the speed of the generation. The expression $100 - C_X$ assess the fault coverage concerning the fault coverage using fault model Y.
The formula (3) can be rearranged into the following formula (4) having single nominator and single denominator:

$$Q_X = \frac{(C_X - C_R) \times m_{XY}}{(101 - C_X) \times (m_X + \ln(T_X) \times m_{XY})} \quad . \tag{4}$$

The disadvantage of the proposed formula is that the calculated value *Qx* has no upper limit. This value is relative. Therefore, it is suitable to compare different fault models in the boundaries of single circuit only.

We carry out the experiments in two rounds. During the first round, we select the best two models among the proposed black delay fault models. We have decided to choose the two best fault models rather than single one since their assessment criterion is constructed in an intuitive way. For the experiment, we use the limited number of the circuits. During the second round, we use the selected black box delay fault models and all available for us circuits.

In the next section, we present the results of the experiment to compare the proposed black delay fault models. The goal of the experiment is to choose the best two fault models and carry out the further experiments.

5. Initial Assessment of the Black Box Delay Fault Models

Bearing in mind the notation introduced in the previous section, we denote by model *X* the proposed fault models M1-M9; the common model *Y* for comparison is the functional delay (FD) fault model [5]. We carried out the experiments on the circuits from the benchmark suite ITC'99 [22]. Because of very large amount of experiments two small (b10 and b11) and one of middle scale b14 circuits were chosen. The models of the benchmark circuits are written in C programming language. We used a Windows machine with 3.2 GHz processor for experiments.

First, certain number of test sequences was randomly generated for every circuit (set *T*). In our case, every randomly generated test sequence starts with an initialization pattern having set signal *Reset*. Next, the number of detected FD was measured for the whole test set composed of separate test sequences. The results are presented in 0. The first column holds the circuit name. The second column shows the number of test sequences. The column under name "Length" reports the length of test sequence. The column under name "k_{YY}" provides the number of detected FD faults. The last column reports the number of selected test sequences according to the fault model FD.

Table 2. Number of detected FD faults on test set T

Name	No	Length	k_{YY}	m_Y
b10	10000	30	335	50
b11	10000	50	758	65
b14	10000	20	15090	1363

Afterwards, all nine proposed black-box fault models were used for the test selection on the same set *T* of test sequences. The results of the experiments are presented in 0-0 for the circuits' b10, b11, and b14, respectively. The structure of all tables is the same. The first column holds the name of the fault model. The second column provides the number of detected faults (k_X) according to the fault model. The column under name "m_X" presents the number of selected test sequences. The column under name "T_X" shows the test selection time. The column under name "k_{XY}" presents the number of detected FD faults. The coverage of detected FD faults is provided in the column under name "C_X". The column under name "m_{XY}" shows the number of test sequences that detect FD faults. The column under name "k_{RY}" presents the number of FD faults that are detected by the first k_X test sequences from initial test set *T*. The coverage of FD faults detected by the first k_X test sequences from initial test set *T* is provided in the column under name "C_R". The last column presents the value of our analytical expression multiplied by 100 in order to easier read the obtained values.

Table 3. Application of fault models M1-M9. Circuit b10

Fault model	k_X	m_X	T_X (sec.)	k_{XY}	C_X (%)	m_{XY}	k_{RY}	C_R (%)	Q_X*100
M1	109899	1037	4	317	94,63	38	319	95,22	-0,32
M2	6576	101	5	308	91,94	32	298	88,96	6,91
M3	31089	4483	119	334	99,70	46	327	97,61	1,57
M4	58642	2514	119	335	100,00	42	325	97,01	4,62
M5	3588	551	122	329	98,21	37	314	93,73	8,15
M6	4804	201	125	324	96,72	33	312	93,13	7,69
M7	3461	1166	120	331	98,81	41	321	95,82	4,09
M8	4870	537	119	329	98,21	37	314	93,73	8,31
M9	53107	1715	1	328	97,91	42	325	97,01	0,71

We should notice that the higher value of criterion Q_X implies the better fault model. Surprisingly, we have gotten negative value of criterion Q_X for the fault model M1. The negative value was obtained since the number m_X of random test sequences from the beginning of test set allows obtaining the higher fault coverage than the same number m_X of the selected test sequences according to the fault model M1. Such a result seems quite reasonable; if the coverage of the test sequences selected according to the fault model cannot surpass the coverage of random test sequences, such a fault model has no value. All the other fault models have gotten a positive assessment. The model M8 is the best one. The model M5 has gotten quite close assessment value to the value of the model M8. These models have the same or similar numbers in all the other columns, as well. The rows of the table for these models are marked in light yellow color. Next, we notice that the other factor that plays quite important role in the quality assessment of the fault models is the number of the selected test sequences (m_X). These two fault models have quite a moderate number of selected test sequences.

Table 4. Application of fault models M1-M9. Circuit b11

Fault model	k_X	m_X	T_X (sec.)	k_{XY}	C_X (%)	m_{XY}	k_{RY}	C_R (%)	Q_X*100
M1	201250	2051	7	756	99,74	63	754	99,47	0,61
M2	7308	142	7	706	93,14	48	699	92,22	2,40
M3	69721	8261	333	758	100,00	65	758	100,00	0,00
M4	179367	4366	335	758	100,00	65	758	100,00	0,00
M5	7106	776	338	740	97,63	62	735	96,97	1,07
M6	7654	111	335	699	92,22	38	687	90,63	2,07
M7	6818	2074	337	755	99,60	63	754	99,47	0,24
M8	7824	367	338	733	96,70	61	719	94,85	3,64
M9	102757	5889	2	758	100,00	64	757	99,87	0,14

For the circuit b11, the fault model M8 is the best one again. However, there is no the other fault model that would have the value of criterion Q_X close to the value of the model M8. The next one is the fault model M2. The rows of the table for these models are marked in light yellow color. Moreover, two fault models have gotten values 0 of the criterion Q_X. This happened since the fault coverage of the random and selected test sequences is 100%. Next, the second factor in the quality assessment of the fault models is the number of the selected test sequences (m_X). These two fault models have the smallest number of the selected test sequences, except the model M2 that is the third one.

Table 5. Application of fault models M1-M9. Circuit b14

Fault model	k_X	m_X	T_X (sec.)	k_{XY}	C_X (%)	m_{XY}	k_{RY}	C_R (%)	Q_X*100
M1	883396	1311	58	13562	89,87	690	13079	86,67	4,799
M2	84507	253	60	11433	75,77	243	10495	69,55	4,797
M3	182534	7057	2227	14791	98,02	1152	14757	97,79	0,74
M4	322830	6055	2228	14602	96,77	1047	14583	96,64	0,30
M5	38896	1753	2193	14436	95,67	822	13371	88,61	13,43
M6	60640	1085	2567	13973	92,60	643	12753	84,51	10,23
M7	40033	3087	2260	14727	97,59	1043	13950	92,45	14,16
M8	63343	2146	2547	14512	96,17	1020	13578	89,98	13,05
M9	406584	949	30	13356	88,51	574	12521	82,98	8,76

For the circuit b14, the fault model M7 is the best one. The fault model M5, which was the second best one for the circuit b10, is ranked into the second place again. The rows of the table for these models are marked in light yellow color. The best model M8 for the circuit b10 and b11 is now ranked into the third place. The value of criterion Q_X for the fault model M8 is quite close to the value of the criterion Q_X for the fault models M7 and M5. This time, the first factor in the quality assessment of the fault models is not quite easily distinguished. Contrary to the circuits' b10 and b11, the second factor in the quality assessment of the fault models even impossible to highlight.

We have already mentioned that the quality assessment of the proposed fault models according to the criterion Q_X is possible in the boundaries of the single circuit only. In order to generalize the results of the experiments we propose to rank the fault models. The base for ranking should be the obtained value of criterion Q_X for the particular circuit. Therefore, we propose the following rules for ranking of the fault models:

1. We assign 7 points for the fault model that obtained the highest value of criterion Q_X for the particular circuit.
2. We assign 6 points for the fault model that obtained the next to highest value of criterion Q_X for the particular circuit and we proceed in such a way in the decreasing order.
3. We do not assign the ranking points for the fault models that were the two last ones according to the value of criterion Q_X for the particular circuit.

The results of ranking are provided in 0. The table contains five columns. The first column holds the names of the fault models. The next three columns provide the ranking points of fault models for the circuits' b10, b11, and b14, respectively. The last column reports the total value of ranking points.

Table 6. Ranking of fault models

Fault model	b10	b11	b14	Total
M1	0	3	2	5
M2	4	6	1	11
M3	1	0	0	1
M4	3	0	0	3
M5	6	4	6	16
M6	5	5	4	14
M7	2	2	7	11
M8	7	7	5	19
M9	0	1	3	4

According to the results of ranking provided in 0, we conclude that M8 and M5 are the best fault models. We choose these fault models for the further experiments. The rows of the table for these models are marked in light yellow color.

In the next section, we present the results of the experiment to select test sequences for the sequential non-scan circuits using the proposed black box delay fault models.

6. Test Selection Based on Black Box Delay Fault Models

We carried out the experiments on the circuits of benchmark suite ITC’99. Two black-box delay fault models M5 and M8 were used in the experiments. We have added to this set of fault models the functional delay fault model FD. This fault model was used for the comparison purposes only. Every test generation lasted 12 hours.

The results of the experiment are presented in 0. The first column holds the name of the circuit. The three rows are devoted for every circuit in the table since the three fault models were investigated. The second column under name “Fault model” provides the name of the fault model. The column under name “No of generated” presents the number of randomly generated test sequences. The larger value in the column means the higher speed of test sequence selection according to the particular fault model. The column under name “No of selected” shows the number of selected test sequences from the generated ones according to the particular fault model. Then, these test sequences were fault simulated. Automatic test pattern generation (ATPG) system TetraMAX was used for transition fault simulation. This ATPG system enables the selection of test sequences that detect transition faults. The column under name “No of selected after fault simulation” presents the number of test sequences that detect the transition faults. The redundancy of selected test sequences, which is obtained by dividing the value of “No of selected” by the value of “No of selected after fault simulation”, is provided in the column “Redundancy”. The transition fault coverage of selected test sequences is shown in the column under name “Transition FC (%) of selected”. The last column presents the transition fault coverage of the number of randomly generated test sequences, which is equal to the number of selected test sequences.

Table 7. Results of test selection. FM – Fault model, NG – No of generated, NS – No of selected, NSA – No of selected after fault simulation, R – redundancy, TFCS – Transition FC (%) of selected, TFCR – Transition FC (%) of random.

Circuit	FM	NG	NS	NSA	R	TFCS	TFCR
b14	FD	136496	2675	1049	2,55	76,52	66,90
	M8	78838	4684	1250	3,75	76,82	71,68
	M5	70566	7496	1379	5,44	76,85	74,11
b15	FD	76598	869	499	1,74	21,80	13,39
	M8	56991	3116	569	5,48	21,12	15,36
	M5	59459	4594	638	7,20	20,41	15,91
b17	FD	7488	439	330	1,33	8,50	6,06
	M8	18057	1796	533	3,37	8,98	7,70
	M5	16228	2363	571	4,14	8,87	7,81
b20	FD	43241	3599	1356	2,65	56,43	49,47
	M8	40981	3240	1161	2,79	52,65	48,33
	M5	42135	4451	1329	3,35	53,68	51,57
b22	FD	14287	3141	1381	2,27	53,10	49,51
	M8	23964	2854	1257	2,27	51,94	48,59
	M5	28851	4394	1561	2,81	53,35	52,38

The analysis of the results provided in 0 reveals that no fault model is the best one when the single criterion for comparison is the transition fault coverage. The fault model FD is the best one for the circuits' b15 and b20. The fault model M5 is the best one for the circuits' b14 and b22. The fault model M8 is the best one for the circuit b17. According to the results of redundancy of test sequences, the fault model FD is the best one, meanwhile the fault model M5 is the worst one. Moreover, the fault coverage of test sequences selected according to the fault model M5 is less productive than the fault coverage selected according to other two fault models in comparison with the same number of random test sequences (see Fig. 2). In Fig.2, we have depicted the difference of the fault coverage of the selected test sequences and the fault coverage of the same number of randomly generated test sequences. The values of difference were calculated using the values in the last two columns of 0. Fig. 2 quite clearly reveals the suitability of the compared delay fault models. The delay fault model FD is the best one. However, the black box delay fault model M8 loses insignificantly to the fault model FD, especially it is true for the largest circuit b22. Meanwhile, the black box delay fault model M5 is the least applicable one.

In general, we conclude that the test sequences for non-scan sequential circuits can be selected according to the black box fault models in the initial stage of the design. We appoint the preference to the fault model M8, which is based on the fault simulation, considers the impact of the transition at the primary inputs to the change of the value at the primary outputs after *p* clock periods.

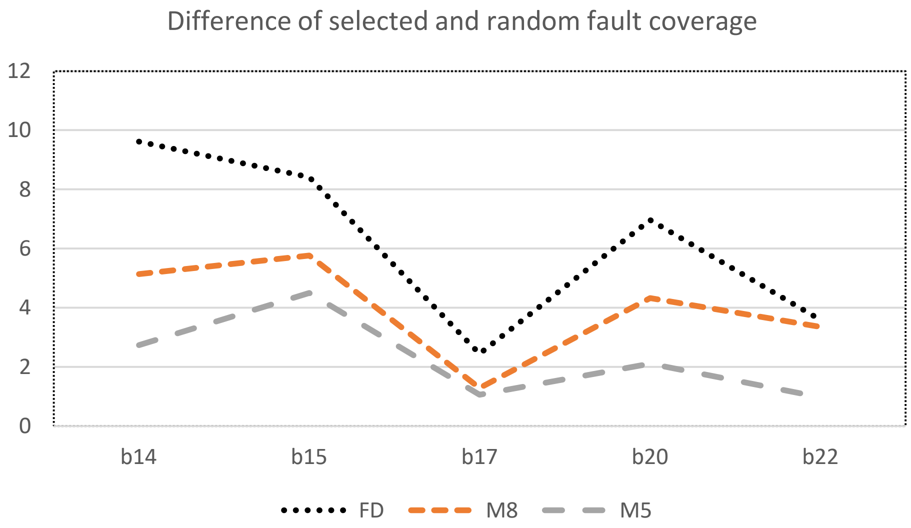

Fig. 2. Difference of fault coverage of selected and random test sequences

7. Conclusions

Functional test generation usually refers to the functional fault models at a high level of abstraction, when primary inputs, primary outputs and state bits are available at least. We presented the black box delay fault models for non-scan sequential circuits at the functional level in order to select the test sequences from randomly generated ones, when the primary inputs and primary outputs are available only. Each test sequence starts on the initialization test pattern. We devised nine black box delay fault models. The obtained results of the selection of the test sequences cannot be compared directly on the base of fault coverage, since the fault models represent the different fault sets. We examined the suggested fault models in two stages. The goal of the first stage was to select the best ones in order to use them in the subsequent investigation. Since the fault sets of proposed fault models are different, we have chosen a single model as the reference one for comparison. This is a functional delay fault model FD. In order to take into account a versatility of the suggested fault models, we constructed criterion that is based on the following three factors: fault coverage, number of the test sequences, and time of the selection. According to our construction, the most important factor is the fault coverage of the selected test sequences compared to the fault coverage of the same number of randomly generated ones. If the selected test sequences cannot obtain the larger fault overage than the same number of the random ones, they have no value. The least important factor is the time of the test selection. During the first stage of the experiment, we carried out the investigation on the three circuits of ITC’99 benchmark suite.

According to our criterion, in order to be for sure we selected two best black box delay fault models, namely M8 and M5. The names were assigned symbolically according to the order of the presentation of the fault models. The model M8 obtained the larger total value of the criterion. Then, we proceeded to the second stage.

During the second stage, we used the three following fault models: functional delay fault model FD, black box delay fault model M8 and black box delay fault model M5. The second stage of the experiment was carried out on the largest circuits of the benchmark ITC'99. Now, the comparison of fault coverage was carried out at the transition level of the circuits. The obtained results of the fault coverage at the transition level demonstrate that the test sequences selected on the base of the black box fault models without knowing directly the inner state of the circuit loses quite insignificantly in comparison with functional delay fault model that uses the inner state of the circuit.

In general, the obtained results allow to conclude that the test sequences for non-scan sequential circuits can be selected according to the black box fault models when the state of the circuit is unavailable. We appoint the preference to the fault model M8 that is based on the fault simulation, considers the impact of the transition at the primary inputs to the change of the value at the primary outputs after *p* clock periods.

References

1. Xiang, D; Chen, M-J; Sun, J-G; Fujiwara H. Improving test effectiveness of scan-based BIST by scan chain partitioning. IEEE Transactions On Computer-Aided Design of Integrated Circuits And Systems, Vol. 24, Iss. 6, 2005, p. 916-927.
2. Kang, J., Seth, S. C., Gangaram V.: Efficient RTL Coverage Metric for Functional Test Selection, Proc. of the 25th IEEE VLSI Test Symposium, 2007, p. 318-324.
3. Bareiša, E., Jusas, V., Motiejūnas, K., Šeinauskas, R.: Functional Delay Clock Fault Models, Information Technology and Control, Vol. 37, No. 1, 2008, 12-18.
4. Bareiša E., Jusas V., Motiejūnas L., Šeinauskas R.: Generating Functional Delay Fault Tests for Non-Scan Sequential Circuits, Information Technology and Control, 2010, Vol. 39, No 2, 100-107.
5. Bareisa E., Jusas V., Motiejunas K., Seinauskas R.: Functional Fault Models for Non-Scan Sequential Circuits, Microelectronics Reliability, 2011, Vol. 51, Issue 12, 2402-2411.
6. Bareisa E., Jusas V., Motiejunas K., Seinauskas R.: Functional Delay Test Generation Approach Using a Software Prototype of the Circuit, Computer Science and Information Systems, 2013, Vol. 10, No. 3, 1165-1184.
7. Sauer, M., Kupferschmid, S., Czutro, A., Polian I., Reddy S., Becker B.: Functional Test of Small-Delay Faults using SAT and Craig Interpolation. In: Proceedings of 43rd IEEE International Test Conference (ITC), 2012, paper 6.3.
8. Dasgupta S., Walther R.G., Williams T.W, Eichelberger E.B., An Enhancement to LSSD and Some Applications of LSSD in Reliability, Availability and Serviceability, Proceedings of the International Symposium on Fault Tolerant Computing, 1981, 880-885.
9. Patil S., Savir J., Skewed-Load Transition Test: Part II, Coverage, Proceedings of the IEEE International Test Conference, 1992, 714-722.
10. Savir J., Patil S., Broad-Side Delay Test, IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems, Vol. 13, No. 8, 1994, 1057-1064.
11. Pomeranz, I., Parvathala, P.K, Patil, S.: Estimating the Fault Coverage of Functional Test Sequences without Fault Simulation. In: Proceedings of 16th IEEE Asian Test Symposium, 2007, 25-30.

12. Fang, H., Chakrabarty, K., Jas, A., Patil, S., Tirumurti, C.: RT-Level Deviation-Based Grading of Functional Test Sequences. In: Proceedings of 27th IEEE VLSI Test Symposium, 2009, 264-269.
13. Fang, H., Chakrabarty, K., Jas, A., Patil, S., Tirumurti, C.: Functional Test-Sequence Grading at Register-Transfer Level, IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 20, Iss. 10, 2012, 1890-1894.
14. Vinutha, K.R., Singh, V., Matrosova, A., Gaur, M.S.: Fault Grading using Instruction-Execution Graph. In: Proceedings of 2010 East-West Design & Test Symposium (EWDTS), 2010, 350-357.
15. Gent, K.; Hsiao, M. S. A Control Path Aware Metric for Grading Functional Test Vectors. Proceedings of 17th IEEE Latin-American Test Symposium (LATS), April 06-08, 2016, p. 51-56.
16. Pomeranz, I.: Concatenation of Functional Test Subsequences for Improved Fault Coverage and Reduced Test Length. IEEE Transactions on Computers, Vo. 61, Iss. 6, 2012, 899-904.
17. Pomeranz, I.: Selection of Functional Test Sequences with Overlaps. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 33, Iss. 7, 2014, 1095-1099.
18. Pomeranz, I. A Generalized Definition of Unnecessary Test Vectors in Functional Test Sequences. ACM Transactions on Design Automation of Electronic Systems, Vol., 20, Iss. 2, 2015, Article Number 29.
19. Last, M; Eyal, S; Kandel, A. Effective black-box testing with genetic algorithms. In: Proceedings of 1st International Haifa Verification Conference, 2006, 134-148.
20. Vos, T. E. J.; Lindlar, F. F.; Wilmes, B. Windisch, A., Baars, A. I., Kruse, P. M., Gross, H., Wegener, J. Evolutionary functional black-box testing in an industrial setting. Software Quality Journal, Vol. 21, Iss. 2, 2013, 259-288.
21. Fraser, G.; Walkinshaw, N. Assessing and generating test sets in terms of behavioural adequacy Software Testing Verification & Reliability. Vol. 25, Iss. 8, 2015, p. 749-780.
22. Corno F., Sonza Reorda M., Squillero G.: RT-Level ITC'99 Benchmarks and First ATPG Results, IEEE Design and Test of Computers, Vol. 17, No. 3, 2000, p. 44-53.

Eduardas Bareisa graduated from Kaunas Polytechnic Institute (Lithuania) in 1975. He received the D.Sc. degree from Kaunas Polytechnic Institute in 1987. Since 2005, he is professor at Department of Software, Kaunas University of Technology, Lithuania. He is author and co-author of more than 70 papers. His research interests include high-level synthesis, VLSI test generation, and software testing.

Vacius Jusas graduated from Kaunas Polytechnic Institute (Lithuania) in 1982. He received the D.Sc. degree from Kaunas Polytechnic Institute in 1988. Since 2006, he is professor at Department of Software, Kaunas University of Technology, Lithuania. He is author and co-author of more than 90 papers. He is Editor of journal “Information Technology and Control”. His research interests include testing of VLSI circuits, design for testability, and design verification.

Kestutis Motiejunas graduated from Dresden University of Technology in 1973 (Germany). He received the D.Sc. degree from Kaunas Polytechnic Institute in 1981 (Lithuania). Since 2006, he is professor at Department of Software, Kaunas University of Technology, Lithuania. He is author and co-author of more than 80 papers. His research interest include testing of VLSI circuits, design for testability, and design verification..

Liudas Motiejunas graduated from Kaunas University of Technology in 2000. He received master degree in Informatics Engineering in 2002 and PhD degree Informatics Engineering in 2007 in Kaunas University of Technology. Since 2011 he is associate professor at Department of Multimedia Engineering, Kaunas University of Technology. His research interests include business rules, business rules engines and image processing. Corresponding author, tel. +370 614 95049.

Rimantas Seinauskas graduated from Kaunas Polytechnic Institute (Lithuania) in 1968. He received the D.Sc. degree from Kaunas Polytechnic Institute in 1972 (Lithuania), and the Habil. D.Sc. degree from St. Petersburg Institute of Fine Mechanics and Optics (Russia) in 1982. Since 1985, he is professor at Department of Software, Kaunas University of Technology, Lithuania. He is author and co-author of more than 100 papers. His research interest include testing of VLSI circuits, design for testability, and design verification.

Received: November 18, 2016; Accepted: June 20, 2017.