

BLACK-BOX FAULT MODELS

Eduardas Bareiša, Vacius Jusas, Kęstutis Motiejūnas, Rimantas Šeinauskas

*Software Engineering Department, Kaunas University of Technology
Studentų 50-406, LT-51368 Kaunas, Lithuania*

Abstract. Four black-box fault models are introduced in the paper. The test generation task for the black-box model is more complicated, because possible realizations of the design must be taken into account. However, the time required to generate tests is not very critical factor, because the test generation can be done in parallel with the circuit synthesis process without a prolongation of Time-to-Market. All the proposed fault models were analyzed and investigated experimentally. On the basis of these results, an appropriate fault model responding to the complexity of the problem being solved can be selected.

1. Introduction

Functional-level test generation must be performed when a gate-level description of the Circuit-Under-Test (CUT) is not available or does not accurately describe the circuit, as is often the case in embedded core designs with Intellectual Property considerations. A test set generated at the functional level is independent and effective for any implementation and, therefore, can be generated at early stages of the design process [1, 2]. Functional Automatic Test Pattern Generation (ATPG) can also be used to identify testability problems before an implementation is selected. Ideally, it is needed a design, a test methodology and fault models that may enable a high-level design validation, the testability enhancement and the test generation in such a way that a high Defects Coverage (DC) would be achieved. Of course, reaching this goal independently from the structural synthesis and from the manufacturing technology is impossible, as the test quality will always depend on the final physical realization. However, significant steps in this direction can be made.

High-level fault models have been proposed for the realization-independent functional testing at RTL level in [3-5]. RTL fault models and quality metrics have been considered in [3]. Logic/arithmetic operations, constant/ variable switch, null statements, if/else, case, for instructions have been considered as RTL fault models. In some cases, their effectiveness in covering stuck-at faults on the circuit's structural description has been ascertained. However, this does not guarantee their effectiveness to uncover physical defects or stuck-at faults. The high-level fault models taken from the software testing have three main advantages: they are well known and quite standardized; they require little calculations, apart from the

complete fault-free simulation; and they are already embedded in some commercial tools. However, while such metrics may be useful to validate the correctness of a design, they are usually inadequate to foresee the gate-level fault coverage with a high degree of accuracy. The observability enhanced statement coverage metric [4, 5] is one of the most used fault models. This fault model requires all statements in the VHDL description to be executed at least once, and their effects to be propagated to at least one primary output.

Some approaches rely on a direct examination of the HDL description [6] or exploit the knowledge of the gate-level implementation [7]. Some extract of the corresponding control machine [8, 9] from a behavioral description is used. The listed approaches are of limited generality and the adequacy of testing defects or of the coverage stuck-at faults on the gate level are not proved.

Functional delay fault models are proposed in [10-12]. The Underwood et al. [10] fault model results in test sets of practical sizes; but its coverage of path delay faults in an arbitrary gate-level implementation of the circuit is low. The Pomeranz and Reddy [11] model results in test sets that cover all the path delay faults in an arbitrary gate-level implementation of the circuit. The main disadvantage of the Pomeranz and Reddy [11] model is that it results in test sets of very large size. A compromise that results in fewer tests at the cost of reduced fault coverage is mentioned in Pomeranz and Reddy [12]. The functional fault model proposed here encompasses the Underwood et al. [10] and Pomeranz and Reddy [11] models in an attempt to combine their advantages. An advantage of functional ATPG for path delay faults over structural ATPG can be seen by considering the number of targeted faults. For structural ATPG, the number of faults is

proportional to the number of paths in the circuit, which is very often exponential in circuit size. In the case of functional ATPG, the number of targeted faults is only proportional to the product of the number of inputs and the number of outputs in the circuit [2].

The behavioral or black-box models were introduced in [1, 13]. The behavioral view represents the system by defining the behavior of its outputs according to the values applied on the inputs without the knowledge of its internal organization. In this case, only the input-output relationship can be determined [13]. The black-box fault models are approximate enough comparing to the other fault models. However, the experimental investigation of these fault models demonstrated high fault coverage on a gate level for the benchmark circuits [1, 13].

Four black-box fault models will be analyzed in this paper.

2. The input-output pin pair fault model

Various fault models based on input-output paths testing were suggested in [1, 13]. In this section, we provide a different representation of the main concepts. Let the circuit have a set of inputs $X = \{x_1, x_2, \dots, x_i, \dots, x_n\}$ and a set of outputs $Z = \{z_1, z_2, \dots, z_j, \dots, z_m\}$. The pin pair fault model considers the stuck-at-0/1 faults occurring at the circuit boundary, and has a weak correlation with the circuit's physical faults. We write x_i^1 and x_i^0 for the input stuck-at-1/0 faults, and z_j^1 and z_j^0 for the output stuck-at-1/0 faults. There are $2n + 2m$ possible stuck-at faults on the inputs and the outputs. Input and output stuck-at fault pairs (x_i^t, z_j^k) , $t=0,1, k=0,1$ are called pin pair faults (PP). The number of possible pin pair faults of the circuit is at most $4 \cdot n \cdot m$. We denote the set of the pin pair faults by

$$P1 = \{(x_i^t, z_j^k) \mid i=1, \dots, n, j=1, \dots, m, t=0, 1, k=0, 1\}.$$

The test stimulus detects the pin pair fault (x_i^t, z_j^k) of the circuit if this test stimulus detects both the input fault x_i^t , and the output fault z_j^k of the pair on the output z_j of the circuit. Sometimes there exists no electric connection between the input and the output, and the pin pair fault defined by these input and output can't be detected. These faults are untestable. The PP fault (x_i^t, z_j^k) of a circuit is testable if a conventional deterministic test generator finds a test stimulus detecting an input fault x_i^t on the output z_j while the negated values (not t – to the input and not k – to the output) are assigned to the input x_i and to the output z_j . The number of testable PP faults equals to $4 \cdot n \cdot m$ minus the number of untestable PP faults. The relationship ratio demonstrates the relation between the number of testable PP faults and the total number of possible PP faults and is computed as follows:

$$\text{Relationship ratio} = \text{Number of testable PP faults} / 4 \cdot n \cdot m.$$

It is a convenient way to represent the detection of PP faults of the circuit by the relationship matrix $C = \|c_{ij}\|_{2n \times 2m}$. The matrix entry $c_{2i-t, 2j-k}$ corresponds to the PP fault (x_i^t, z_j^k) . We assume that the matrix entry is equal to one when the fault is detected and the entry is equal to zero otherwise. In general, circuits may have a lot of untestable PP faults. The untestable PP faults complicate test generation significantly. The test generator can prove the untestability of the fault only by enumerating all the possibilities in the search space. However, the search space is typically very large even for circuits of moderate size.

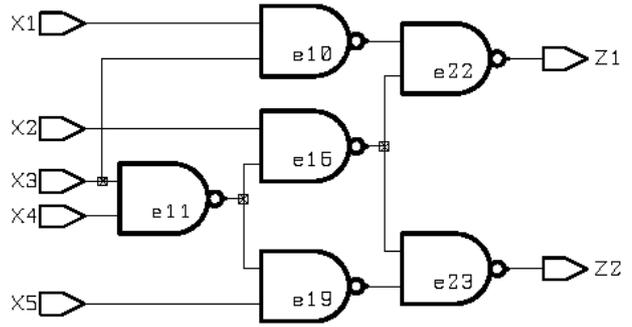


Figure 1. Circuit c17

Four entries of the matrix define relations between the input and the output of the circuit. The existing path with the even number of inverters shows that the faults (x_{x1}^0, z_{z1}^0) , (x_{x1}^1, z_{z1}^1) can be detected. The upper left entry and the lower right entry of the four entries of the matrix correspond to these faults, as it is shown in Table 1. Whereas, PP faults between the input $x1$ and the output $z2$ are untestable, because there exists no path to connect them in the circuit and the corresponding entries of the matrix C have zero values. Observe that if there exist no connecting path for any realization of the circuit between the input and the output, the corresponding PP faults are untestable. However, no one should assert opposite, as the circuit's realization may be redundant with redundant connections. Thus, the existence of an even or odd path (a path having an odd number of inverters) in the circuit does not guarantee the detection of the corresponding PP faults. Yet, the bulk of untestable PP faults can be determined according to the structure of the circuit.

There exist two paths connecting the input $x3$ to the output $z1$; one path has the even number of inverters, the other – the odd number of inverters. Consequently, all the four corresponding entries of the matrix C are equal to one. Whereas, there are four paths connecting the inputs $x3$ and $x4$ to the output $z2$ through the odd number of inverters. Thus, a single detection of a corresponding PP fault does not guarantee the detection of all stuck-at faults in these paths. In general, the detection of the PP fault should guarantee the detection of the stuck-at faults in one of the paths at least. However, one has to remember that the

activation of several paths simultaneously (by one stimulus) may not guarantee that the stuck-at faults of at least one path would be detected.

Table 1. The Matrix C of the Circuit c17

	z1		z2	
x1	1	0	0	0
	0	1	0	0
x2	1	0	1	0
	0	1	0	1
x3	1	1	0	1
	1	1	1	0
x4	0	1	0	1
	1	0	1	0
x5	0	0	1	0
	0	0	0	1

In general, it is not possible to relate the PP fault to the stuck-at faults of the circuit unambiguously, because the PP fault doesn't determine the propagation path of the fault effect. The detection of the pin pair fault (x_i^t, z_j^k) is related to the paths of the circuit between the input i and the output j . The faults (x_i^0, z_j^k) and (x_i^1, z_j^k) are symmetric. If the test stimulus detecting one of them is known, the other fault can be detected using the adjacent test stimulus where the value on the input i is negated. The adjacent stimuli detect both pin pair faults by the same path of the circuit. If both test stimuli, which detect faults, are not adjacent (they differ in more than one value), then both test stimuli may activate different paths between the input i and the output j .

Note that there may be many paths between the input i and the output j . If the pin pair fault is detected only once, it is evident that there may be a lot of paths, which were not activated, after detecting all the pin pair faults; the same could be true also for the stuck-at faults. However, the experimental researches proved that the situation is different. The test stimulus detecting pin pair faults does not detect only one and a half percent of stuck-at faults on the average [14]. This result can be explained by the fact that each test stimulus detects the stuck-at faults on several paths between the input and the output. The stuck-at faults of the gates, which allow the activation of the path, are detected as well. In the case of the circuit c17, the single paths starting on the inputs x3, x4 and passing through the gates e11, e19, and e23 to the output z2 are not possible to activate, and the stuck-at-0 fault on the input of the gate e19 is not detected. However, this fault is detected while detecting the PP faults from the input x5 to the output z2.

The other reason why the functional test possesses a high quality is that every stuck-at fault can be detected by several paths between inputs and outputs. If a stuck-at fault lies on several paths starting on the K inputs and ending at the L outputs, then it can be detected by the activation of $K \times L$ paths. If such a fault

is not detected by one path, then it can be detected by another path out of the $K \times L$ paths.

Next reason why the functional test with no guarantee of activating all paths between inputs and outputs has a high quality is the following: in general, a test stimulus detects several pin pair faults and the set of the test stimuli may detect most of PP faults more than once. A new path can be activated by the detection of the same PP fault using a new test stimulus as well. Consequently, the strategy for improving the quality of a functional test becomes evident: every PP fault has to be detected several times. However, in such a case, the set of the test stimuli is oversized significantly and there is no guarantee that a new path will be activated in the detection of the PP fault next time. Therefore, the improvement of the quality of the functional test by increasing the number of the test stimuli, which detect the PP faults, is not rational. One has to look for such black-box fault models that increase the set of test stimuli moderately and purposefully. Such possible fault models are discussed in the further sections.

Various strategies for obtaining a set of test stimuli that detect PP faults are possible. For every stimulus detecting the PP fault (x_i^0, z_j^k) ((x_i^1, z_j^k)), it is possible to form a stimulus detecting the symmetric PP fault (x_i^1, z_j^k) ((x_i^0, z_j^k)); the other strategy is to strive that every PP fault would be detected by a different stimulus.

It is interesting to know an influence of the compaction of the test set that detects PP faults to the quality and length of the test. Such issues will be discussed preliminary on the test set of circuit c17.

Table 2. Stimuli for the circuit c17

N	x1	x2	x3	x4	x5	z1	z2
1	0	1	0	0	0	1	1
2	0	1	1	0	1	1	1
3	1	0	0	1	1	0	1
4	1	0	0	0	0	0	0
5	0	0	1	0	1	0	1
6	1	0	1	1	0	1	0
7	1	1	1	1	0	1	0
8	0	1	0	1	0	1	1
9	0	1	1	1	0	0	0

The test of 9 stimuli was obtained automatically; this test (Table 2) detects all the PP faults of the circuit c17, but it does not detect the stuck-at-1 at the first input of gate e19 of the given realization. A characteristic feature of this test is that every stimulus of the test sequence detects some PP fault, which was not detected by the previous stimuli in the sequence. Such a test sequence can be redundant. The irredundant test is distinguished by the feature that after eliminating some stimulus from the set of the test, the remaining stimuli would not detect some PP faults. The elimination of 1, 3, and 7 stimuli from the automatically

generated test formed the irredundant test containing six stimuli. The compaction of the test did not harm the coverage of the test at the gate level of the circuit realization presented in Figure 1; the coverage results are the same as before the minimization. The optimized synthesis of circuit *c17* yields the realization that is presented in Figure 2; in the latter case, the irredundant six stimuli of circuit *c17* detect all the stuck-at faults of this realization.

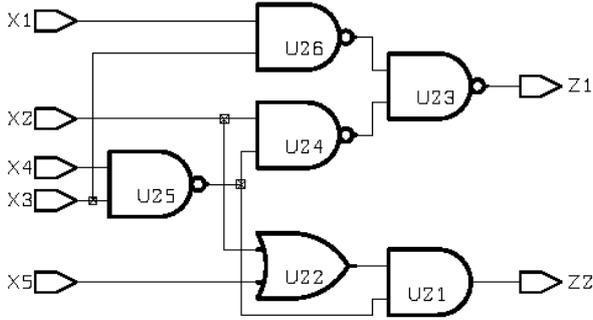


Figure 2. Optimized Circuit *c17*

3. The input-input-output pin triplet fault model

The PP fault model requires the activation of the path between an input and an output at least one time. The activation of the pair of the paths would increase the number of the separate paths activation. The pin triplets possess such property.

The input-input-output pin stuck-at fault triplets (x_i^t, y_h^p, z_j^k) , $t=0,1$, $k=0,1$, $p=0,1$ are called the pin triplet faults (PT) [14]. The number of possible pin triplet faults of the circuit is at most $4 \times n \times n \times m$. The set of the pin triplet faults is denoted as follows:

$$PT = \{(x_i^t, y_h^p, z_j^k) \mid i=1, \dots, n, h=1, \dots, n, j=1, \dots, m, t=0, 1, k=0, 1, p=0, 1\}.$$

The test stimulus detects the pin triplet fault (x_i^t, y_h^p, z_j^k) of the circuit if this test stimulus detects the pin faults x_i^t , y_h^p and z_j^k of the triplet on the output z_j of the circuit. The pin triplet fault requires the activation of two paths from the inputs to the output by the same test stimulus. All possible pairs of the paths activation are considered.

The PT fault covers the PP fault if x_i and y_h is the same input. The pin fault triplet (x_i, x_i, z_j) reduces to the pin fault pair (x_i, z_j) .

Now we analyze the relation of PT faults to the structure of the circuit. Every triplet fault is related to the activation of two paths of circuit from the input x to the output $z(x-z)$ and from the input y to the output $z(y-z)$ by the same test stimulus. The paths may have both even and odd number of inverters and the faults on the paths can be detected when the value on the output is 1 or 0. Different stuck-at faults are detected in case of different values on the output. Generalized cases of PT faults are presented in Table 3.

Table 3. The generalized PT faults

Faults	Comments
(x_i^0, y_h^0, z_j^0)	The activation of two circuit's paths $x-z$ and $y-z$ having an even number of inverters when the value at the output is 1
(x_i^0, y_h^0, z_j^1)	The activation of two circuit's paths $x-z$ and $y-z$ having an odd number of inverters when the value at the output is 0
(x_i^0, y_h^1, z_j^0)	The activation of the path $x-z$ having an even number of inverters and the path $y-z$ having an odd number of inverters when the value at the output is 1
(x_i^0, y_h^1, z_j^1)	The activation of the path $x-z$ having an odd number of inverters and the path $y-z$ having an even number of inverters when the value at the output is 0
(x_i^1, y_h^0, z_j^0)	The activation of the path $x-z$ having an odd number of inverters and the path $y-z$ having an even number of inverters when the value at the output is 1
(x_i^1, y_h^0, z_j^1)	The activation of the path $x-z$ having an even number of inverters and the path $y-z$ having an odd number of inverters when the value at the output is 0
(x_i^1, y_h^1, z_j^0)	The activation of two circuit's paths $x-z$ and $y-z$ having an odd number of inverters when the value at the output is 1
(x_i^1, y_h^1, z_j^1)	The activation of two circuit's paths $x-z$ and $y-z$ having an even number of inverters when the value at the output is 0

It is convenient to represent the detection of PT faults using the ternary matrix $L = \|l_{i,h,j}\|_{2n \times 2n \times 2m}$, where the matrix entry $l_{2i-t, 2h-p, 2j-k}$ corresponds to the PT fault (x_i^t, y_h^p, z_j^k) . We will assume that the entry $l_{2i-t, 2h-p, 2j-k}$ gets value 1 when the fault is detected, otherwise - 0. In general, circuits have many untestable PT faults, which complicate test generation a lot.

The ternary matrix can be presented in the form of two-dimensional matrix layers. The layer corresponds to the value at the output and to detectable fault as well. The stuck-at 0 is detected ($k=0$) when the value at the output is 1, and the stuck-at-1 is detected ($k=1$) when the value at the output is 0. So, two layers correspond to each output of the circuit in the ternary matrix. Four entries of the matrix L are assigned to each pair of inputs in the layer, which corresponds to the output, as it is shown in Table 4. The rows and columns correspond to the inputs of the circuit. Two rows and two columns are used to describe every input of the circuit. They intersect in four entries of the matrix L . The value 1 in the entries of Table 4 indicates the detection of the corresponding PT fault. The entries of the matrix layer will be denoted according to Table 4 when the value at the output is 1 ($z_j=1$), but in this case, z_j^1 is changed to z_j^0 in all the entries.

The entries of the three-dimensional matrix L can be linked to the structure of the circuit. For the left upper entry of four entries of the three-dimensional matrix L the value 1 is assigned if both inputs are

connected to the output by the paths having an even (odd) number of inverters when the value at the output is 0 (1). For the right lower entry the value 1 is assigned if both inputs are connected to the output by the paths having an odd (even) number of inverters when the value at the output is 0 (1). For the left lower entry the value 1 is assigned if the input x_i is connected to the output by a path having an odd (even) number of inverters and the input x_j is connected to the output by a path having an even (odd) number of inverters when the value at the output is 0 (1). Similarly, for the right upper entry the value 1 is assigned if the input x_i is connected to the output by a path having an odd (even) number of inverters and the input x_j is connected to the output by a path having an even (odd) number of inverters when the value at the output is 0 (1).

Table 4. Layer of the Three-Dimensional Matrix L when $z_j=0$

		x_{h-1}	x_h		x_{h+1}	
x_{i-1}						
x_i			(x_i^1, y_h^1, z_j^1)	(x_i^1, y_h^0, z_j^1)		
			(x_i^0, y_h^1, z_j^1)	(x_i^0, y_h^0, z_j^1)		
x_{i+1}						

The entries of the three-dimensional matrix L can be linked to the structure of the circuit. For the left upper entry of four entries of the three-dimensional matrix L the value 1 is assigned if both inputs are connected to the output by the paths having an even (odd) number of inverters when the value at the output is 0(1). For the right lower entry the value 1 is assigned if both inputs are connected to the output by the paths having an odd (even) number of inverters when the value at the output is 0(1). For the left lower entry the value 1 is assigned if the input x_i is connected to the output by a path having an odd (even) number of inverters and the input x_j is connected to the output by a path having an even (odd) number of inverters when the value at the output is 0(1). Similarly, for the right upper entry the value 1 is assigned if the input x_i is connected to the output by a path having an odd (even) number of inverters and the input x_j is connected to the output by a path having an even (odd) number of inverters when the value at the output is 0(1).

If both inputs x_i and y_j are connected to the output z by the paths having either an even or odd number of inverters, then only for one entry – either the left upper one or the right lower one – out of the four entries of the three-dimensional matrix the value 1 can be

assigned. If one of the inputs is connected to the output by a path having an even number of inverters and the other input is connected to the output by a path having an odd number of inverters, then for either the left lower entry or the right upper one the value 1 can be assigned. If one of the inputs is connected to the output by a path having an even number of inverters and the other input is connected to the output by a paths having an even number of inverters as well as by a paths having an odd number of inverters, then for two adjacent entries out of the four ones the value 1 can be assigned. Finally, when both inputs are connected to the output by paths having an even number of inverters as well as by paths having an odd number of inverters, for all four entries the value 1 can be assigned.

Note that the inputs $x1$ and $x2$ of the circuit $c17$ are connected to the output $z1$ by the paths having an even number of inverters. The layers of the three-dimensional matrix of circuit $c17$, when $z1=0$ and $z1=1$, are shown in Table 5 and Table 6, respectively. Two columns and two rows are linked to every input of the circuit. In both tables, only single entry has the value 1 in the intersection of the inputs $x1$ and $x2$. Whereas, the input $x3$ is connected to the output $z1$ by a path having an even number of inverters as well as by a path having an odd number of inverters, consequently, only for two adjacent entries the value 1 is assigned. These entries are symmetrical in the layers corresponding to the values 0 and 1 at the output $z1$. It has to be mentioned that values can be symmetrical placed not only in the entries of different layers of output. Some values are symmetrical in the same layer with respect to the diagonal crossing from the left upper corner to the right lower corner. Tables 5-8 present all the layers of the three-dimensional matrix obtained according to the structure of the circuit $c17$.

Table 5. Layer of the Three-Dimensional Matrix L when $z_1=0$

	$x1$	$x2$	$x3$	$x4$	$x5$
$x1$	1	0	1	0	1
	0	0	0	0	0
$x2$	1	0	1	0	1
	0	0	0	0	0
$x3$	1	0	1	0	1
	1	0	1	0	0
$x4$	0	0	1	0	0
	1	0	1	0	1
$x5$	0	0	0	0	0
	0	0	0	0	0

Table 6. Layer of the Three-Dimensional Matrix L when $z_1=1$

	x1	x2	x3	x4	x5					
x1	0	0	0	0	0	0	0	0	0	
	0	1	0	1	1	1	1	0	0	0
x2	0	0	0	0	0	0	0	0	0	0
	0	1	0	10	1	1	1	1	0	0
x3	0	1	0	1	1	0	1	0	0	0
	0	1	0	1	0	1	1	0	0	0
x4	0	1	1	0	1	1	1	0	0	0
	0	0	1	0	0	0	0	0	0	0
x5	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0

Table 7. Layer of the Three-Dimensional Matrix L when $z_2=0$

	x1	x2	x3	x4	x5					
x1	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
x2	0	0	1	0	1	1	1	1	1	0
	0	0	0	0	0	0	0	0	0	0
x3	0	0	1	0	0	0	0	0	1	0
	0	0	1	0	0	1	0	1	1	0
x4	0	0	1	0	0	0	0	0	1	0
	0	0	1	0	0	1	0	1	1	0
x5	0	0	1	0	1	1	1	1	1	0
	0	0	0	0	0	0	0	0	0	0

Table 8. Layer of the Three-Dimensional Matrix L when $z_2=1$

	x1	x2	x3	x4	x5					
x1	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
x2	0	0	0	0	0	0	0	0	0	0
	0	0	1	0	1	1	1	1	1	0
x3	0	0	1	0	0	1	0	1	1	0
	0	0	1	0	0	0	0	0	1	0
x4	0	0	1	0	0	1	0	1	1	0
	0	0	1	0	0	0	0	0	1	0
x5	0	0	0	0	0	0	0	0	0	0
	0	0	1	0	1	1	1	1	1	0

The test stimulus is defined by the vector $A=\langle a_1, a_2, \dots, a_i, \dots, a_n \rangle$, where $a_i \in \{0,1\}$. The response of the circuit is defined by the vector $B=\langle b_1, b_2, \dots, b_j, \dots, b_m \rangle$, $b_j \in \{0,1\}$. At the beginning, for all entries of the three-dimensional matrix L the value 0 is assigned. The indices of the matrix have the following values: $i^*, h^*=1, 2, 3, \dots, 2n, j^*=1, 2, 3, \dots, 2m$. For the entry of the three-dimensional matrix that is indexed $i^*=2i+a_i, h^*=2h+a_h, j^*=2j+b_j$ the value 1 is assigned, when the complement of the values at the inputs a_i and a_h entails the complement of the value at the output b_j as well.

Till now we considered input/input/output fault models. Input/output/output fault models can be developed in the similar way. We denote the set of the pin triplet faults by

$$PT2=\{(x_i^t, y_h^p, z_j^k) \mid i=1, \dots, n, h=1, \dots, m, j=1, \dots, m, t=0, 1, k=0, 1, p=0, 1\}.$$

The test stimulus detects a pin triplet fault (x_i^t, y_h^p, z_j^k) of the circuit if this test stimulus detects the pin faults x_i^t, y_h^p and z_j^k of the triplet at the outputs z_j and y_h of the circuit. The pin triplet fault requires the activation of two paths that connect input x_i to the outputs y_h and z_j on the same test stimulus. All possible pairs of the activation of the paths are considered.

Every triplet fault is related to the activation of two paths that connect the input x to the output $z(x-z)$ and the input x to the output $y(x-y)$ by the same test stimulus. The paths may have both even and odd number of inverters and can be activated when the value at the output is 1 or 0. Different stuck-at faults are detected in case of different values at the output. The generalized PT2 faults can be presented in the same way as PT faults (see Table 3). For example, consider the PT2 fault (x_i^0, y_h^0, z_j^0) . The detecting of this fault requires the activation of two circuit's paths x_i-y_h and x_i-z_j having an even number of inverters when the signal value at the outputs y_h and z_j and at the input x_i is 1

In the case of PT2 faults, the base for selecting the pairs of paths differs substantially, and the test stimuli obtained according to the PT2 model can differ from the ones, obtained according to the PT model. There is no theoretical background to prove the superiority of one of these models. We used the PT fault model till now, though the comparison of the models PT and PT2 by experiments has not been performed yet.

In general, the PT fault model allows detecting every PP fault several times, because the PT fault consists of the PP fault pairs. This depends on the number of PP faults linked to the same output. Therefore, the test obtained using the PT model should be significantly longer and should test more stuck-at faults, as the test stimuli better cover the paths of the circuit. This conclusion was validated by the experiments presented in [14].

The test, generated for the circuit c17 according to the PT fault model, has 12 test stimuli and detects all stuck-at faults. Despite this fact, we will demonstrate by a simple example why a test for PP and PT faults doesn't guarantee the detection of stuck-at faults at the gate level of the circuit. The circuit is shown in Figure 3. It has one fan-out with branches b_1 and b_2 . Table 9 lists all possible input stimuli in the initial columns. The PP faults, the stuck-at faults on the fan-out branches and the PT faults detectable by the corresponding input stimulus are shown in the following columns of Table 9.

The test generator constructed according to PT or PP fault model will always select the input stimuli 2, 3 and 6, because no other stimuli can detect the corresponding PT or PP faults. The PP fault (b^1, d^1) can

be detected by one of the three stimuli: 1, or 4, or 5. In the case of a selection of the input stimulus 5, all stuck-at faults of the fan-out will be detected. Some stuck-at faults of the fan-out remain undetected in the case of selecting the input vector 1 or 4. Only both stimuli 1 and 4 can guarantee the detection of all stuck-at faults of the fan-out. The test generation according to the PP fault model or the PT fault model can never guarantee the selection of both stimuli 1 and 4 or the selection of the stimulus 5. Note that the adjacent input stimuli can improve the detection of the stuck-at faults. This conclusion will be discussed in the next section.

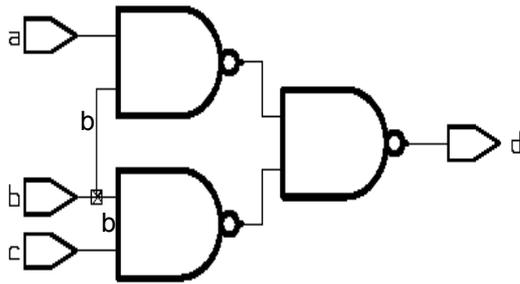


Figure 3. The tricky circuit

Table 9. The stimuli and faults detected by stimuli

N	a	b	c	d	PP faults	Fan-out stuck-at faults	PT faults
0	0	0	0	0			
1	0	0	1	0	(b^1, d^1)	b_2^1	(b^1, b^1, d^1)
2	0	1	0	0	(c^1, d^1) (a^1, d^1)		(a^1, c^1, d^1)
3	0	1	1	1	(b^0, d^0) (c^0, d^0)	b_2^0	(b^0, c^0, d^0)
4	1	0	0	0	(b^1, d^1)	b_1^1	(b^1, b^1, d^1)
5	1	0	1	0	(b^1, d^1)	$b_1^1 b_2^1$	(b^1, b^1, d^1)
6	1	1	0	1	(b^0, d^0) (a^0, d^0)	b_1^0	(a^0, b^0, d^0)
7	1	1	1	1	(b^0, d^0)		

Despite the drawback of the fault models demonstrated above, the test stimuli generated according to the mentioned models for the entire suite of the benchmark circuits detect stuck-at faults of any realization surprisingly well. A few reasons why the results are unexpectedly good can be seen. Firstly, the values at the different outputs depend on the same inputs. A stuck-at fault can be detected on several outputs. The test stimulus for one output can detect stuck-at faults at the other outputs as well. Actually, each test stimulus can detect several PT faults. Consequently, each PT fault may be tested more than once.

4. The model of activating function terms

The relationship matrix is useful not only for representing the detection of PP or PT faults. The

relationship matrix holds information, which could be used to write the variables of the direct and reverse function of the circuit. As it is seen in Table 1, each output variable is linked to two columns. The first column indicates the input variables that have an impact to the value of the reverse function of the output. The second column indicates the input variables that influence the value of the direct function of the output. The output function depends on the input variable if there is at least one value 1 in the intersection of column and rows, where column corresponds to the output variable and rows correspond to the input variable.

Every input variable is linked to two rows. The upper row indicates that the variable is negated in the function. The bottom row indicates that the variable has a direct form. The function terms may have variables both in the direct form and in the negated form. The variable of a term either in the direct or in the negated form will be referred as a literal of the term. Thus, according to the first column of Table 1, the terms of the reverse function of the output z_1 have

the following literals: $\bar{z}_1 = \bar{f}(x_1, \bar{x}_2, \bar{x}_3, x_3, \bar{x}_4)$. According to the second column of Table 1, the terms of the direct function of the output z_1 have the following literals:

$z_1 = f(x_1, x_2, \bar{x}_3, x_3, x_4)$. Similarly, according to the third and the fourth columns of Table 1, the reverse function is $\bar{z}_2 = \bar{f}(x_2, x_3, x_4, \bar{x}_5)$, and the direct function is $z_2 = f(x_2, x_3, x_4, x_5)$.

The relationship matrix, which is presented in Table 1, can be created for each single test stimulus. The relationship matrix for the stimulus $A = \langle a_1, a_2, a_3, a_4, a_5 \rangle = \langle 10100 \rangle$ is presented in Table 10. Note that the value 1 at the output z_1 is determined by the variables x_1 and x_3 . These variables form the activating term $x_1 x_3$. Similarly, the value 0 at the

output z_2 is determined by the values \bar{x}_2 and \bar{x}_5 , which

form the activating term $\bar{x}_2 \bar{x}_5$. The relationship matrix for the next stimulus $\langle a_1, a_2, a_3, a_4, a_5 \rangle = \langle 11110 \rangle$ is presented in Table 11. From this table, the activating term x_1 for the output z_1 and the activating term $x_3 x_4$

for the output z_2 are obtained. The activating functions

$z_1 = x_1 x_3 + \bar{x}_1$ and $z_2 = \bar{x}_2 \bar{x}_5 + x_3 x_4$ can be formed from the obtained activating terms in a similar way as the logical functions are formed. The activating term $x_1 x_3$ includes all literals of the other activating term \bar{x}_1 . The former activating term is more valuable for test generation because it includes more variables. Consequently, the activating term, whose literals are not covered by other activating term, will be called a significant activating term. The activating term x_1 is not a significant one, as its literals are covered by the activating term $x_1 x_3$. The activating functions of the outputs are formed from the significant activating terms only. The significant activating term of the function is considered as the fault model, and the

values of the inputs forming it are considered as the test stimulus that detects this fault. A test stimulus may detect several activating terms of the function of the different outputs. The activating term (AT) fault model has a disadvantage that the number of faults cannot be calculated in advance and thus the usual concepts of test completeness cannot be used. All the activating terms of the function can be determined only if the response of the circuit is found to all possible input stimuli. Often this is not the case for real circuits. Because of this restriction there is no guarantee that the function will have all the possible activating terms. The different sets of activating terms can be obtained for the circuit. This conclusion means that the number of faults is changeable and it depends of their formation order. This fault model is more suitable for simulation-based test generation. In general, the functions of the circuits have many activating terms and these terms can be detected by a large number of test stimuli. The test stimuli obtained according to this fault model detect the stuck-at faults at the gate level better if compared to PP and PT fault models. The test set generated automatically for the circuit C17 according to this fault model has 15 stimuli, which detect all the stuck-at faults of the circuit.

Table 10. The relationship matrix for stimulus <10100>

	z ₁		z ₂	
x ₁	0	0	0	0
	0	1	0	0
x ₂	0	0	1	0
	0	0	0	0
x ₃	0	0	0	0
	0	1	0	0
x ₄	0	0	0	0
	0	0	0	0
x ₅	0	0	1	0
	0	0	0	0

Table 11. The relationship matrix for stimulus <11110>

	z ₁		z ₂	
x ₁	0	0	0	0
	0	1	0	0
x ₂	0	0	0	0
	0	0	0	0
x ₃	0	0	0	0
	0	0	1	0
x ₄	0	0	0	0
	0	0	1	0
x ₅	0	0	0	0
	0	0	0	0

Now we will discuss the correlation between the activating terms of the function constructed according to the relationship matrix and the usual terms of the logical function. Direct and reverse logical functions

of the outputs of the circuit C17 are defined as follows:

$$z_1 = x_1x_3 + x_2\bar{x}_4 + x_2\bar{x}_3, \quad \bar{z}_1 = \bar{x}_1\bar{x}_2 + \bar{x}_1x_3x_4 + \bar{x}_2\bar{x}_3,$$

$$z_2 = x_3x_5 + \bar{x}_4x_5 + x_2\bar{x}_4 + x_2\bar{x}_3, \quad \bar{z}_2 = \bar{x}_2\bar{x}_5 + x_3x_4.$$

Note that the terms of the reverse logical function of the output z₂ are the same as those obtained according to the relationship matrix. All the significant activating terms, which were obtained automatically for this example, coincide with the terms of the logical functions. However, such a result cannot be obtained in all the cases. Let us see the example, where the

function is defined as follows: $f = x_1x_2x_3 + \bar{x}_1\bar{x}_2\bar{x}_3$, $\bar{f} = x_1$

$x_2 + \bar{x}_1\bar{x}_3 + x_2\bar{x}_3$. The significant activating terms of the

reverse function constructed according to the

relationship matrix are the following: $\bar{f} = x_1 + \bar{x}_1 + x_2 +$

$x_2 + x_3 + x_3$ (Table 12); they differ significantly from the

terms of the reverse logical function.

Table 12. Input stimuli and activating terms

	x1	x2	x3	y	Terms
1	0	0	0	1	$\bar{x}_1\bar{x}_2\bar{x}_3$
2	0	0	1	0	x ₃
3	0	1	0	0	x ₂
4	0	1	1	0	\bar{x}_1
5	1	0	0	0	x ₁
6	1	0	1	0	\bar{x}_2
7	1	1	0	0	\bar{x}_3
8	1	1	1	1	x ₁ x ₂ x ₃

Usually, the term of the logical function includes not all the variables. It is possible to assign arbitrary values to the variables that are not included in the term. Such an assignment does not have impact on the value of the logical function. The direct function of output the z₁ of circuit C17 includes the term x₁x₃. Such a term indicates that the values 1 at the inputs x₁ and x₃ determine the value 1 at the output z₁ independently the values at the inputs x₂, x₄, x₅. This rule is not valid for the activating terms that are obtained according to the relationship matrix. The reverse function includes the term x₁, which determines the value 0 at the output. However, if the values at the inputs x₂ and x₃ are 1, the value at the output is 1 as well. But this drawback does not hinder to use the fault model of the significant activating terms for test generation.

Note that on the base of the partially obtained activating terms there is a possibility to generate the input stimuli that would reveal new terms. We will not present the algorithm but the ideas to achieve this result we will explain on the base of the example. Let us have two significant activating terms of the func-

tion $z_1 = x_1x_3 + x_2\bar{x}_4$. It is known that the function has more activating terms. On the base of the current

activating terms, it is possible to generate input stimuli, where every term of the function includes at least one literal that has the value 0. The literal x_1 is equal to 0 when the value at the input x_1 is 0, whereas the literal \bar{x}_4 is equal to 0 when the value at the input x_4 is 1. According to these rules, the 4 input stimuli are obtained that define the following values for the literals of the terms: 01+01, 01+10, 10+01, 10+10. Using these values, the following input stimuli can be created: $\langle x_1, x_2, x_3, x_4 \rangle = \langle 0010 \rangle, \langle 0111 \rangle, \langle 1000 \rangle, \langle 1101 \rangle$. These 4 input stimuli determine all the significant activating terms of the reverse function $\bar{z}_1 = x_1 \bar{x}_2 + \bar{x}_1 x_3 x_4 + \bar{x}_2 \bar{x}_3$ and insignificant term \bar{x}_2 . This example demonstrates that the properties of activating terms allow using them in the generation of new terms.

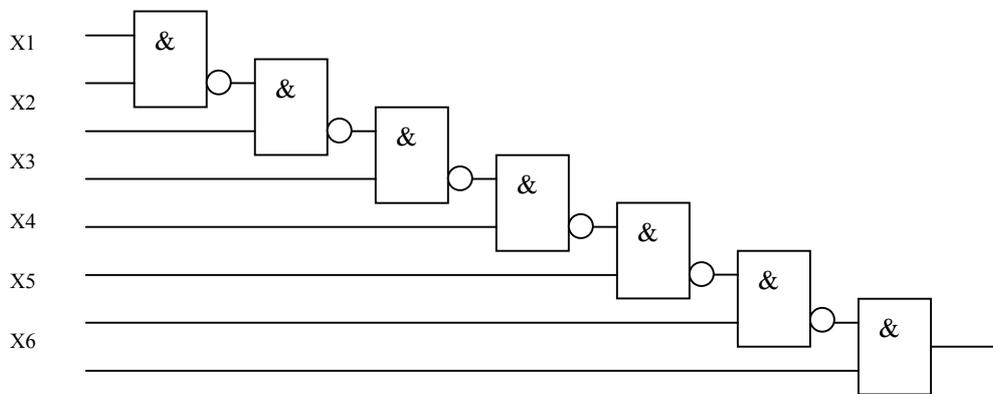


Figure 4. Example circuit

However, the functional fault models do not refer to the structure of the circuit. We now discuss the possibilities to evaluate the length of the paths in the black-box model of the circuit. Let's analyze the circuit presented in Figure 4.

After applying the stimulus that consists of all 1's, the next stimulus is the same but the value at the input x_1 is 0. The transition of the value is propagated to the output of the circuit. The transition enables detection of all the stuck-at faults and transition delay faults of the path. The transition can propagate to the output if the values at the off-path inputs allow this transition. The longer path is the more off-path inputs have to allow the transition to propagate on the path. The input is significant to the propagation path if the complement of the value at the input stops the propagation of transition on the path. In our example, the values at the inputs x_2-x_8 are significant to propagation path from the input x_1 to the output. The propagation path from the input x_2 to output is maintained by the same seven input values as well. The transition at the input x_3 can be maintained by five or six input values already. This example illustrates that the shorter is the path in the circuit, the less values at the input can be assigned. In the real circuits, which

5. The fault models based on the propagation path

Every test stimulus has the active inputs; the flipping the values at these inputs entails the change of the values at some outputs. The pair of stimuli that differ in the value of the active input only activates the path from the active input to the output. Such a pair of stimuli is called the adjacent stimuli. The signal transition at the input allows detecting the stuck-at-0 and stuck-at-1 faults. The adjacent stimuli detect transition delay faults as well. However, our experience with the detection of all PP faults by adjacent stimuli shows that about 5 percent of the transition delay faults on the average are not detected [15]. In order to reduce the average percent of the undetected transition delay faults the PP faults should be detected through the longest paths of the circuit.

have many fan-outs and fan-ins, it is possible that the longer path would require less maintaining values, but that is not true in general case. The longer propagation path requires more values at the inputs that allow the propagation of the transition.

In general, the pair of the input and output (x_i, z_j) can be characterized by the number s of inputs that maintain the propagation of the transition, and we denote the corresponding functional fault model by $(x_i, z_j)^s$. In order to detect this fault, the activation of the path from the input x_i to the output z_j is not enough, but the stimulus, which activates this path, must have the most maintaining values at the inputs. In this case, we can consider the question regarding the quality of the fault detection. The larger is the value s , the higher is the quality of the test stimulus. The purpose of test generation is not only to check all the faults, but to obtain the highest quality of test stimuli as well.

In order to estimate the influence of the quality of the test stimuli to the detection of transition delay faults, test stimuli that have the highest and the lowest quality for Benchmark ISCAS'85 circuits were generated. The lowest quality means that the fault $(x_i, z_j)^s$ was detected by the test stimulus that has the

smallest value s , i.e. the smallest number of the maintaining inputs. The comparison showed surprising results in some cases. The test stimuli of the lowest quality achieved better results of transition delay faults detection for some circuits. The careful analysis revealed that the pairs of the test stimuli of the highest quality were very similar. Perhaps, this fact influenced the results. However, the pairs of the test stimuli of the highest and the lowest quality detect the transition faults better (by 1.5 percent approximately) than a double set of test stimuli, which does not evaluate the quality of the stimuli. This result may be explained by the fact that the test stimuli of the highest and the lowest quality activate different paths between the input and the output. According to this fact, it may be purposeful of evaluating not only the number of the maintaining inputs, but the values at these inputs as well. The inputs, which maintain the propagation path, may be considered as the literals of the activating term. Thus, the input/output pair (x_i, z_j) is related to the activating term. The input/output pair (x_i, z_j) may be related to many activating terms. One can expect that the propagation of the transition from the input to the output, when the maintaining terms are different, will cover more different paths at the gate level. This result should influence the coverage of the detection of transition faults positively, and transition faults should be detected through the longer paths as well. However, such researches are not performed yet.

6. Conclusions

Four black-box fault models were introduced and analyzed. The application of these models produces the test sets of different size. The larger set of tests is distinguished by better quality of fault coverage. All the proposed fault models were analyzed and researched on examples. On the basis of presented considerations, an appropriate fault model responding to the complexity of the problem being solved can be selected.

In general, the test generation task for the black-box model is more complicated, because possible realizations of the design must be taken into account. Therefore, the test set for the black-box model is larger as compared to the test set for a particular realization of the circuit. However, the time taken by the test generation is not so critical, because the test generation can be done in parallel with the circuit synthesis process without a prolongation of Time-to-Market. Large test sets for the black-box model can be compacted by performing analysis not only according to the stuck-at faults, but also according to various defects for a particular realization.

References

- [1] V. Jusas, K. Paulikas, R. Seinauskas. Procedures for Selection of Validation Vectors on the Algorithm

- Level. 2nd IEEE Latin-American Test Workshop, February 11-14, 2001, Cancun, Mexico, 90-95.
- [2] M. Michael, S. Tragoudas. ATPG Tools for Delay Faults at the Functional Level. *ACM Transactions on Design Automation of Electronic Systems*, Vol.7, No.1, January 2002, 33–57.
- [3] M.B. Santos, F.M. Goncalves, I.C. Teixeira, J.P. Teixeira. RTL-Function Test Generation for High Defects Coverage in Digital SOCs. *Proc. of the IEEE European Test Workshop, ETW'00*, 2000, 99-104.
- [4] S. Devadas, A. Ghosh, K. Keutzer. An Observability-Based Code Coverage Metric for Function Simulation. *Proceedings/ACM international Conference on Computer Aided Design*, 1996, 418-424.
- [5] F. Fallah, S. Devadas, K. Keutzer. OCCOM: Efficient Computation of Observability-Based Code Coverage for Functional Verification. *Proceedings 35th Design Automation Conference*, 1998, 152-157.
- [6] F. Ferrandi, F. Fummi, D. Sciuto. Implicit test Generation for Behavioral VHDL Models. *Proceedings IEEE International Test Conference*. 1998, 587-596.
- [7] E. M. Rudnick, R. Vietti, A. Ellis, F. Corno, P. Pri-netto, M. Sonza Reorda. Fast sequential circuit test Generation Using High-Level and Gate-Level Techniques. *Proceedings IEEE European Design Automation and Test Conference*, 1998, 570-576.
- [8] K.-T. Cheng, A.S. Krishnakumar. Automatic Generation of Functional Vectors Using the Extended Finite State Machine Model. *ACM Trans. On Design Automation of Electronic Systems*, Vol.1, No.1, Jan. 1996, 57-79.
- [9] D. Moundanos, J.A. Abraham, Y.V. Hoskote. A Unified Framework for Design Validation and Manufacturing Test. *Proceedings IEEE International Test Conference*, 1996, 875-884.
- [10] B. Underwood, W.O. Law, S. Kang, H. Konuk. Fastpath: A path-delay test generator for standard scan designs. *Proceedings of 1994 International Test Conference*, 1994, 154–163.
- [11] I. Pomeranz, S.M. Reddy. On testing delay faults in macro-based combinational circuits. *Proceedings of 1994 International Conference on Computer-Aided-Design*, 1994, 332–339.
- [12] I. Pomeranz, S.M. Reddy. Functional test generation for delay faults in combinational circuits. *Proceedings of 1995 International Conference on Computer-Aided-Design*, 1995, 687–694.
- [13] R. Seinauskas, E. Bareisa. Test Selection Based on the Evaluation of Input Stuck-at Faults Transmissions to Output. *Information technology and control, Kaunas, Technologija*, 1996, No.2(3), 15-18.
- [14] E. Bareisa, V. Jusas, K. Motiejūnas, R. Seinauskas. The Realization-Independent Testing Based on the Black Box Fault Models, *Informatica*, Vol.16, No.1, 2005, 19-36.
- [15] E. Bareisa, V. Jusas, K. Motiejūnas, R. Seinauskas. Application of Functional Delay Tests for Testing of Transition Faults and Vice Versa. *Information Technology and Control, Kaunas, Technologija*, 2005, Vol.34, No.2, 95 - 101.

Received January 2006.