ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2009, Vol. 38, No. 3

ON THE ENRICHMENT OF FUNCTIONAL DELAY FAULT TESTS

Eduardas Bareiša, Vacius Jusas, Kęstutis Motiejūnas, Rimantas Šeinauskas

Software Engineering Department, Kaunas University of Technology
Studentų St. 50-406, LT–51368 Kaunas, Lithuania
e-mail: kestas@soften.ktu.lt

Abstract. The testing phase is becoming the most crucial part of the overall design process, which delays the time-to-market of the digital devices. In order to reduce the complexity of test generation and to decrease the time-to-market, one needs to begin the test design at higher levels of abstraction. In this paper a new approach for functional delay test enrichment is described. The test enrichment procedure does not increase the test size and is fast because it does not require test generation. The described approach enriches the test patterns using fault simulation. The performed experiments demonstrate effectiveness of the proposed approach.

1. Introduction

The complexity of digital devices is growing continuously. The test design requires more time and is becoming the most crucial part of overall design process that delays the time-to-market of the digital device. Many recent system-on-a-chip (SoC) integrated circuits incorporate pre-designed and reusable components, variously referred to as intellectual property (IP) circuits or cores. Such circuits are frequently supplied by third-party vendors and are extremely hard to test when embedded in a SoC because their functions are specified only in high-level terms. This is done either to protect the circuits' IP content or else to allow system designers to synthesize their own low-level (gate-level) implementations. IP components pose new challenges for testing.

Test generation is developed in two directions. The usual trend is when the test is generated for the circuit at the structural level. In this case, the main problem is the test generation time, because it directly influences the time-to-market. The task of test generation is quite complicated, especially for sequential circuits. Therefore the design for testability is applied for such circuits. This helps to reduce the cost of test development. But the scan design allows a synchronous sequential circuit to be brought to states that the circuit cannot reach during functional operation. As a result, it allows the circuit to be tested using test patterns that are not applicable during functional operation. This leads to unnecessary yield loss.

The other important direction of test generation is the functional test development at high level of abstraction. In the initial stages of the design, the structural implementation of the design is not known. Therefore the task of the test generation is more complex, because the test has to be generated for all the possible implementations. But the test development can be accomplished in parallel with other design stages. In this case, the time of test generation is not a critical issue. During design process the software prototype of the circuit is created according to the specification. The software prototype simulates the functions of the circuit, i.e. enables to calculate the output values according to the input values. The functional test can be generated on the base of the software prototype. The test patterns generated in such a way can be used for the verification purposes as well. If the generation of the functional test encounters some difficulties, in order to facilitate the task of test generation the state variables of the software prototype can be used as the primary inputs and the primary outputs. In such a case, the generated test can be applied only for the scan designed circuit, but the correspondence between the state variables and the flip-flops of the scan register has to be established.

The functional test is based on the function of the circuit, which can be designed in many ways. The possible defects of the circuit depend on the implementation. The test is usually developed according to the specific implementation and it is designed to detect the defects of this particular implementation. The manufacturing test can be developed only on the base of the specific implementation. Meanwhile, the functional test is not related to the particular implementation because it is generated from a circuit's specification rather than its gate level implementation. The implementation independence of functional test has several advantages over implementation-dependent test. The functional test can be used to correct testability problems early in the design process [1], to

identify the design errors [2], to test many potential implementations [3-5], and to detect hard-to-detect faults at the gate level implementation [4, 6]. Speed binning and power sorting is typically done with functional test patterns [7, 8] as well as interface region around the IP circuits in SOC is a good candidate for high-level delay testing because its implementation details are not known to either the SOC or IP designers [9].

One of the approaches for deriving tests to achieve high defect coverage is based on the generation of n–detection tests. An *n*-detection test is one where each fault is detected either by n different tests, or by the maximum number of different tests that can detect the fault if this number is smaller than n. Butler and Mercer [10] stress that *n*-detection of target faults leads to significant single detection of non-target faults, regardless of the level of abstraction in which the circuit and the faults are described.

In this work, we describe a postprocessing procedure for functional delay test enrichment. The procedure targets a test pattern set T for functional delay faults. It modifies the test patterns in T so that the number of detections of functional delay faults is increased.

The remainder of this paper is organized as follows: in Section 2 we review the related work. Section 3 presents the procedure of functional delay test enrichment. Section 4 presents experimental results, and Section 5 concludes this paper.

2. Related Work

The possibilities of test quality enhancement using various approaches are analyzed in [11-25].

Neophytou, Michael and Tragoudas [11] presented a method to enrich the transition-fault test with additional test patterns having certain property. This property is related to the paths sensitized by the test pattern. The idea is to add a number of new test patterns to the set such that they still detect the targeted transition faults but through paths that have not been sensitized by the original test pattern set. The generated test pattern sets have higher quality since events propagate through many critical paths and thus are more likely to detect a delay violation in the circuit. It is expected that, on average, this procedure will allow the number of paths along which the event propagates to increase proportionally to the number of test patterns per transition fault [11].

To improve the quality of tests for path delay faults, an m-tuple test generation procedure for path delay faults is described in [12]. Under an m-tuple test, each m-tuple of target faults is detected by at least one test pattern. An m-tuple test has advantages similar to an *n*-detection test in that it results in several test patterns for every target path delay fault p, thus increasing the likelihood of testing p under worst-case delay conditions. In addition, it increases the likelihood of accidentally detecting non-target path delay faults [12].

Another possibility of test quality enhancement called sensitivity of adjacent input patterns was proposed in [13]. Sensitive adjacent input vectors can be generated for each test pattern of the test set. Since a change in the value of a single input of sensitive adjacent input vectors changes the output vector, it is likely that the presence of a fault on a path from a sensitive input to a sensitive output will be detected. Generated sensitive adjacent input vectors are likely to be sensitive to the presence of a defect, and are likely to result in higher fault coverage [13].

Bareiša et al. [14] suggested complementing the existent test suites of the IP core with all sensitive adjacent patterns or with a subset of them. Then the suitable test patterns for the synthesized gate level implementation have to be selected on the base of the fault simulation. The experiment presented in [14] proves that such a complement enhances the test quality for any synthesized IP core gate level description. Bareiša et al. [14] point out that the application of sensitive adjacent patterns is a cheap way to adopt test patterns for the re-synthesized gate level description of IP core, because the fault simulation is not so critical task as test generation.

A deterministic procedure of adjacent stimuli generation was suggested in [15]. It is based on the assumption that input stimuli, which are similar to test patterns, have good testing features. The search among such input stimuli improves the overall efficiency. It is evaluated that the adjacent stimuli generation allowed improving the efficiency of random search up to 30%. Consequently, it is suggested the integrated application of random and adjacent stimuli generation during functional test design process [15].

The generation of *n*-detection tests and their capabilities in detecting untargeted faults and defects were studied in [16-20]. An *n*-detection test detects each target fault n times, by n different test patterns. By increasing the number of detections of target faults, *n*-detection test generation for $n > 1$ increases the likelihood of detecting untargeted faults and defects. Generation of an *n*-detection test requires repeated applications of a test generation process to target faults that are not yet detected *n* times. Each time a fault is targeted, a different test pattern must be generated for it. This increases the complexity of test generation.

A procedure for forming *n*-detection tests without applying a test generation procedure to target faults is described in [21]. The proposed procedure accepts a one-detection test. It extracts test cubes for target faults from one-detection test and then merges the cubes in different ways to obtain an *n*-detection test. Merging of cubes does not require test generation. Fault simulation is required for extracting test cubes for target faults [21].

N-detection may lead to large tests where many test patterns do not help increase the defect coverage [22]. The problem of control of test size increment was considered in [22-24].

Pomeranz and Reddy [22] introduced variable *n*-detection tests where different target faults are targeted different number of times. In a variable n-detection test, only selected faults are targeted n times. Other faults are targeted between *1* and *n-1* times. The motivation for introducing variable *n*-detection tests was to control the size of test pattern set as n was increased. The number of times each fault is targeted is determined by a parameter that measures the usefulness of multiple test patterns for the fault in detecting defects. This parameter is based on the number of paths through the fault site [22]. The use of variable number of fault detections while transforming the pin pair test into functional delay test was suggested in [23]. The performed experiments show the effectiveness of this proposal. The restriction of the number of fault detections allowed shortening the test size almost twice [23].

In the paper [24] three parameters of an *n*-detection test to measure the saturation of the test generation process were defined: 1) the fraction of faults detected n times or less by the test; 2) the fraction of faults detected fewer than n times by the test; and 3) the test set size relative to the size of a one detection test. Based on these parameters and the rationale for computing *n*-detection tests, Pomeranz and Reddy [24] defined saturation to occur at the value of n where one of the three parameters reaches a threshold specified for it. The thresholds were selected based on experimental results.

The main drawback of all reviewed techniques is the test size increment. Another disadvantage of almost all mentioned approaches lies in the use of test generation for enrichment of test pattern sets.

A. Bareiša et al. in [25] introduced an approach of enrichment of static functional tests that doesn't possess the disadvantages mentioned above. In the next section we present the adaptation of the approach proposed in [25] for enrichment of dynamic functional tests.

3. A procedure of enrichment of functional delay fault test

We consider the test sets that are generated for detection of functional delay faults. A test for the functional delay fault is a pair of input patterns <u, v> that propagates a transition from a primary input to a primary output of a circuit [26]. Under functional delay fault model proposed in [26], a fault is a tuple (I, O, t I, t O), where I is an input of the circuit under test (CUT), O is a CUT output, t I is a rising or falling transition at I, and t O is a rising or falling transition at O. Thus, four functional delay faults are related with every input/output (I/O) pair and the total number of faults is $4 \times n \times m$, where n is the number of inputs of the CUT and m is the number of outputs of the CUT.

```
procedure EoFDT
    INPUT:    circuit C, corresponding functional delay fault set F, test pattern
              pair set T, number of iterations It
    OUTPUT:   enriched test pattern pair set T^E
1.  l=1
2.  repeat
3.      for each T_k ∈ T do
4.          determine F_k
5.          for i=1 to n do
6.              if (t^1_{k,i} ≠ t^2_{k,i}) then
7.                  set the same signal value: t^1_{k,i} = t^2_{k,i} (t^2_{k,i} = t^1_{k,i})
8.              else
9.                  set signal value transition: t^1_{k,i} =NOT(t^1_{k,i}) (t^2_{k,i} =NOT(t^2_{k,i}))
10.             determine F^*_k
11.             if F_k ⊆ F^*_k then
12.                 F_k=F^*_k
13.             else
14.                 restore signal values t^1_{k,i} and t^2_{k,i}
15.         end
16.     end
17.     l=l+1
18. until k > It
end procedure
```

Figure 1. The pseudocode of procedure EoFDT

The pseudocode of the procedure for Enrichment of Functional Delay Test (procedure EoFDT) is shown in Figure 1. In the rest of the paper the following notations will be used:

- T – the set of test pattern pairs;
- T_k – the test pattern pair ($T_k \in T$)

- $t^1_{k,i}$ – the signal value (1 or 0) of the first pattern of the test pattern pair T_k on circuit input i;
- $t^2_{k,i}$ – the signal value (1 or 0) of the second pattern of the test pattern pair T_k on circuit input i;
- F – the set of functional delay faults of particular circuit;
- F_k – the set of functional delay faults that are detected on test pattern pair T_k;
- F^*_k – the set of functional delay faults that are detected on the modified test pattern pair T_k
- n – the number of circuit inputs.

The procedure EoFDT modifies each test pattern pair T_k of the set T in such way that the modified test pattern pair T^*_k detects all functional delay faults detectable on the initial pattern pair T_k and, probably, some additional functional delay faults not detectable on T_k. Therefore, the enriched test pattern set T^E may detect some functional delay faults that are not detectable on the test pattern set T or, at least, increases the number of detections of some functional delay faults. Modification of the test pattern pair bases on in [27] proved lemma and its corollaries. In paper [27], the necessary conditions are defined for function-robustly propagation of signal value transition from the considered circuit input to the circuit output. Thus all functional delay faults are detected on the enriched test pattern set T^E in function-robustly manner.

Further we provide a short explanation of lines 6-9 of procedure EoFDT. There are four possibilities of modification of test pattern pair T_k. Namely, if the signal values are not equal in test patterns of T_k on circuit input i ($t^1_{k,i} \neq t^2_{k,i}$), we can change signal value either of the first test pattern or of the second one. In case when the signal values are equal ($t^1_{k,i} = t^2_{k,i}$), we have the same choice as well. Therefore, the four different operation modes of procedure EoFDT are as follows:

1. if $t^1_{k,i} \neq t^2_{k,i}$ then $t^2_{k,i} = t^1_{k,i}$ and if $t^1_{k,i} = t^2_{k,i}$ then $t^1_{k,i}$ =NOT($t^1_{k,i}$), let's name this Mode M_1_1;
2. if $t^1_{k,i} \neq t^2_{k,i}$ then $t^2_{k,i} = t^1_{k,i}$ and if $t^1_{k,i} = t^2_{k,i}$ then $t^2_{k,i}$ =NOT($t^2_{k,i}$) (Mode M_1_2);
3. if $t^1_{k,i} \neq t^2_{k,i}$ then $t^1_{k,i} = t^2_{k,i}$ and if $t^1_{k,i} = t^2_{k,i}$ then $t^1_{k,i}$ =NOT($t^1_{k,i}$) (Mode M_2_1);
4. if $t^1_{k,i} \neq t^2_{k,i}$ then $t^1_{k,i} = t^2_{k,i}$ and if $t^1_{k,i} = t^2_{k,i}$ then $t^2_{k,i}$ =NOT($t^2_{k,i}$) (Mode M_2_2).

The most prominent features of the proposed functional test enrichment procedure are: 1) procedure EoFDT does not expand the initial test pattern set T, i.e. there is no test size addition; 2) procedure EoFDT does not require test generation. The described approach enriches the test patterns using functional delay fault simulation. Thus, the computing time of the procedure EoFDT depends linearly on the test size.

A.Bareiša et al. in [25] showed that their test enrichment procedure can be incorporated into test generation system and then used as dynamic test enrichment procedure or it can be applied for relaxation of test pattern sets. The procedure EoFDT can be employed for these purposes as well. When procedure EoFDT is used for dynamic test enrichment, it processes each test pattern pair T_k before T_k is included into test pattern set T. In this case, many faults not yet detectable on the generated test pattern set T may be detected, particularly at the beginning of the test generation. The outcome of dynamic test enrichment should be the reduction of test size.

The procedure EoFDT can be easily modified to use it for the relaxation of test pattern sets. The relaxation of test patterns means changing, where it is possible, of fully specified test pattern bits into unspecified (don't care) bits. The modification affects Lines 6-14 of the procedure EoFDT. These lines have to be changed to:

6. $t^2_{k,i}$ =NOT($t^2_{k,i}$);
7. determine F^*_k;
8. if $F_k = F^*_k$ then;
9. $t^1_{k,i}$ = "unspecified"; $t^2_{k,i}$ = "unspecified";
10. else;
11. $t^2_{k,i}$ =NOT($t^2_{k,i}$).

Moreover, there is needed only one iteration. The motivation behind relaxing of test pattern sets is to make these tests amendable to addressing additional issues beyond detection of the targeted faults [19]. For example, the unspecified bits can be specified in such manner that power dissipation during test application is minimized or the unspecified bits can be specified appropriately to detect additional faults. Such flexible tests are important in various compression schemes for on-chip or off-chip test embedding [19].

4. Experimental results

In this section we present results of the application of the proposed test enrichment procedure to ISCAS'85 benchmark circuits.

The test pattern sets for functional delay faults were generated for the black box model of the circuits [28]. The black box model represents a system by defining the behaviour of its outputs according to the values applied to its inputs without the knowledge of its internal organization. The black box models written in the C programming language were used by the test generation for the functional delay faults.

We used three various types of functional delay tests for the experiments. First tests, let's name them T1, are single-input transition (SIT) tests designed using deterministic test generation technique [29]. Other two tests T2 and T3 are multi-input transition (MIT) tests. The tests T2 are obtained using random test generation techniques where each test pattern of the pair <u, v> is generated randomly, and tests T3 are constructed using deterministic test generation approach [29]. The initial test pattern sets T1 and T2

ensure 100% coverage of targeted functional delay faults, whereas the test pattern sets T3 are incomplete.

The results of test pattern set enrichment are reported in Tables 1-4. In Tables 1 and 2, after the circuit name we show the number of detected functional delay faults (DFDF), and the average number of functional delay faults detections under T, computed as follows. The sum of detections of all functional delay faults is divided by the number of detected functional delay faults. Next for each of four test enrichment modes, we show the average number of detections under enriched test pattern set T^E, the improvement of the average number of detections, expressed in percent, and the number of iterations. Table 3 contains the data of the same kind except column 2 which is omitted. The best results are shown in bold.

Table 1. Results of complete SIT test pattern set enrichment

Circuit	DFDF	Av. T1	Mode M_1_1			Mode M_1_2			Mode M_2_1			Mode M_2_1		
			Av. $T1^E$	Imp. in %	It.	Av. $T1^E$	Imp. in %	It.	Av. $T1^E$	Imp. in %	It.	Av. $T1^E$	Imp. in %	It.
c432	540	1.9	**9.2**	**394**	3	3.1	65	2	9.1	388	3	3.1	67	3
c499	5184	1.8	**34.9**	**1874**	4	2.0	16	2	**34.9**	**1874**	4	2.0	16	1
c880	1326	2.0	**20.7**	**914**	5	15.4	652	4	20.3	895	4	15.4	654	5
c1355	5184	1.8	**35.7**	**1925**	4	1.9	8	1	35.7	1925	4	1.9	8	1
c1908	3004	4.3	**17.5**	**311**	3	6.2	45	2	17.5	310	3	6.2	45	3
c2670	3320	1.9	**36.0**	**1777**	5	20.1	950	4	35.9	1774	4	20.3	958	4
c3540	2588	2.9	**13.3**	**352**	4	6.2	111	3	13.3	351	4	6.2	111	3
c5315	10540	4.2	46.6	1008	5	46.8	1014	3	46.4	1004	5	**46.9**	**1016**	3
c6288	3068	3.9	**5.1**	**30**	2	5.0	26	2	**5.1**	**30**	2	5.0	26	2
c7552	12188	5.4	**36.2**	**573**	4	26.2	387	3	36.1	572	3	26.2	387	3
Aver.	4694	3.0	**25.5**	**916**	3.9	13.3	327	2.6	25.4	912	3.6	13.3	329	2.8

Table 2. Results of complete MIT test pattern set enrichment

Circuit	DFDF	Av. T2	Mode M_1_1			Mode M_1_2			Mode M_2_1			Mode M_2_1		
			Av. $T2^E$	Imp. in %	It.	Av. $T2^E$	Imp. in %	It.	Av. $T2^E$	Imp. in %	It.	Av. $T2^E$	Imp. in %	It.
c432	540	4.7	**8.4**	**79.3**	4	6.4	37.3	3	7.6	61.8	3	6.8	44.6	4
c499	5184	10.1	**10.3**	**2.4**	4	10.2	0.7	1	10.3	2.0	4	10.2	0.9	2
c880	1326	11.8	**26.9**	**127.2**	4	25.3	114.1	3	22.3	88.3	3	25.5	115.8	4
c1355	5184	9.5	**9.6**	**1.3**	4	9.5	0.4	1	9.6	1.2	4	9.6	0.6	2
c1908	3004	15.6	**30.6**	**96.8**	3	18.6	19.3	2	29.4	89.1	3	18.6	19.8	3
c2670	3320	11.6	**29.4**	**153.1**	3	24.7	113.0	3	26.5	128.1	3	24.6	111.6	3
c3540	2588	10.6	18.6	75.6	4	15.5	46.5	3	**19.0**	**79.0**	3	16.6	57.1	3
c5315	10540	26.3	**55.2**	**109.7**	4	49.2	86.6	4	52.8	100.4	4	51.2	94.5	3
c6288	3068	17.5	18.1	3.3	2	17.9	2.1	1	**18.5**	**5.5**	2	18.3	4.3	1
c7552	12188	45.5	**65.7**	**44.2**	3	57.5	26.2	4	59.9	31.7	3	56.8	24.9	4
Aver.	4694	16.3	**27.3**	**69.3**	3.5	23.5	44.6	2.5	25.6	58.7	3.2	23.8	47.4	2.9

Table 3. Results of incomplete MIT test pattern set enrichment

Circuit	Av. T3	Mode M_1_1			Mode M_1_2			Mode M_2_1			Mode M_2_1		
		Av. $T3^E$	Imp. in %	It.	Av. $T3^E$	Imp. in %	It.	Av. $T3^E$	Imp. in %	It.	Av. $T3^E$	Imp. in %	It.
c432	3.9	**5.0**	**26.3**	3	4.7	20.0	2	4.1	5.0	2	4.8	22.2	3
c499	10.07	10.15	0.7	2	10.13	0.6	1	10.11	0.4	1	**10.16**	**0.8**	2
c880	10.1	**19.8**	**96.5**	4	19.7	96.0	3	15.9	57.7	3	**19.8**	**96.5**	4
c1355	9.51	9.55	0.4	2	9.54	0.3	1	9.55	0.4	1	**9.56**	**0.5**	2
c1908	16.5	**20.2**	**22.8**	3	17.8	8.2	2	18.9	14.9	3	17.6	7.0	3
c2670	15.1	**24.5**	**62.4**	3	23.1	53.2	2	20.4	35.5	3	22.7	50.2	3
c3540	10.0	13.5	35.0	3	13.4	34.0	3	14.0	39.8	3	**14.6**	**45.6**	3
c5315	30.0	**44.1**	**47.0**	3	37.1	23.8	3	41.2	37.4	4	39.4	31.4	4
c6288	14.40	14.41	0.1	1	14.41	0.1	1	**14.74**	**2.4**	1	**14.74**	**2.4**	1
c7552	44.6	**59.0**	**32.3**	3	20.3	25.6	4	52.4	17.4	3	52.8	18.4	5
Aver.	16.4	**22.0**	**32.3**	2.7	20.3	25.6	2.2	20.1	21.1	2.4	20.6	27.5	3.0

The following points can be seen from Tables 1-3. The procedure EoFDT was able in all cases to enrich the test pattern sets generated for detection of functional delay faults, i.e. for all circuits, in all operation modes and for all types of tests. The application of procedure EoFDT produced the best outcome for complete SIT test pattern sets T1 where the improvement of the average number of detections ranges from 327% (Mode M_1_2) to 916% (Mode M_1_1) on average. The enrichment of MIT test pattern sets T2 and T3 brings far worse outcome: the best result of 69.9% was got for complete MIT test pattern sets T2 in operation Mode M_1_1 and the worst improvement (20.3%) of the average number of detections is in operation Mode M_1_2 for incomplete MIT test pattern sets T3. However, there is simple explanation of such big difference in improvement of the average number of detections between SIT and MIT tests. Namely, if we consider not the improvement but only average number of detections we can see that this number is very low (3.0, on average) for initial SIT test pattern sets T1. Whereas the initial MIT pattern test sets T2 and T3 possess much higher average number of detections (T2 – 16.3, T3 – 16.4, on average). After application of procedure EoFDT, the average numbers of detections become comparable, and actually the enriched MIT test pattern sets T2 display the best outcome in ratings.

Now let's consider the efficiency of separate operation modes of procedure EoFDT. The procedure EoFDT converges after maximum five iterations. Normally, four iterations are needed in test enrichment Mode M_1_1, three iterations are enough almost in all cases for other modes. The Mode M_1_1 showed outstanding outcome for all types of analyzed functional delay test pattern sets. Therefore, if the primary goal of functional delay test enrichment is improving of test quality in detecting of functional level faults we have to prefer the operation Mode M_1_1 of procedure EoFDT.

In Section 3, it was predicted that the enriched test pattern set T^E may detect some functional delay faults that are not detectable on the initial test pattern set T. The experimental results of the enrichment of incomplete test pattern set T3 reported in Table 4 support this assumption. In Table 4, after the circuit name we show the number of detectable functional delay faults, the number of functional delay faults detected on initial test pattern set T3, and the functional delay fault coverage of T3 expressed in percent. Next, for each of four test enrichment modes, we show the number of functional delay faults detected on enriched test pattern set $T3^E$, the functional delay fault coverage of $T3^E$ expressed in percent, and the improvement of the functional delay fault coverage. The best results are shown in bold as well.

Table 4. Functional delay fault coverages of T3 and $T3^E$

Circuit	FD	DFDF T3	Cov.T3	Mode M_1_1			Mode M_1_2			Mode M_2_1			Mode M_2_1		
				DFDF $T3^E$	Cov. $T3^E$	Imp.	DFDF $T3^E$	Cov. $T3^E$	Imp.	DFDF $T3^E$	Cov. $T3^E$	Imp.	DFDF $T3^E$	Cov. $T3^E$	Imp.
c432	540	438	81.11	462	85.56	4.44	447	82.78	1.67	**463**	**85.74**	**4.63**	448	82.96	1.85
c499	5184	5177	99.86	**5180**	**99.92**	**0.06**	**5180**	**99.92**	**0.06**	5178	99.88	0.02	**5180**	**99.92**	**0.06**
c880	1326	1115	84.09	1250	94.27	10.18	1245	93.89	9.80	**1260**	**95.02**	**10.94**	1253	94.49	10.41
c1355	5184	5174	99.81	**5182**	**99.96**	**0.15**	5179	99.90	0.10	5176	99.85	0.04	5180	99.92	0.12
c1908	3004	2409	80.19	2540	84.55	4.36	2518	83.82	3.63	2524	84.02	3.83	**2559**	**85.19**	**4.99**
c2670	3320	2311	69.61	2393	72.08	2.47	2393	72.08	2.47	**2416**	**72.77**	**3.16**	2415	72.74	3.13
c3540	2588	2223	85.90	**2332**	**90.11**	**4.21**	2308	89.18	3.28	2330	90.03	4.13	2324	89.80	3.90
c5315	10540	8077	76.63	**8255**	**78.32**	**1.69**	8135	77.18	0.55	8248	78.25	1.62	8208	77.87	1.24
c6288	3068	2887	94.10	2888	94.13	0.03	2888	94.13	0.03	**2903**	**94.62**	**0.52**	**2903**	**94.62**	**0.52**
c7552	12188	11188	91.80	**11422**	**93.72**	**1.92**	11327	92.94	1.14	11418	93.68	1.89	11323	92.90	1.11
Aver.	4694	4100	86.31	4190	89.26	2.95	4162	88.58	2.27	**4192**	**89.39**	**3.08**	4179	89.04	2.73

The application of procedure EoFDT increased the coverage of functional delay faults in all cases. The best outcome we got for circuit c880 where the improvement of functional delay fault coverage is approximately 10% in all operation modes. The average improvement ranges from 2.27% (Mode M_1_2) to 3.08% (Mode M_2_1). The operation Mode M_1_1 displays the second best result, namely 2.95%, which is very close to the outcome of operation Mode M_1_2. Thus the conclusion, that the operation Mode M_1_1 of procedure EoFDT is preferable for improving of test quality in detecting of functional level faults, stays valid.

Many authors state that the *n*-detection tests are effective in detecting untargeted faults and defects [16-20]. In order to examine the influence of the improvement of the average number of detections on detecting untargeted faults, we will use the surrogate fault approach described in [30]. A surrogate fault model consists of faults that are not directly targeted by the test generation procedure being used. Under this approach, the test set is generated for a target fault model and simulated on the surrogate faults. The surrogate fault coverage is taken as an indication of the untargeted defect coverage achieved by the test set. The premise behind the use of surrogate faults is that

if test sets T_x and T_y are generated for a fault model M, and if T_x achieves a higher coverage than T_y of faults belonging to a surrogate fault model M_{Surr}, which is different from the target fault model M, then T_x also achieves a higher defect coverage than T_y [30]. We will use transition faults as surrogates for untargeted defects.

The results of transition fault simulation are reported in Tables 5-7, after the circuit name we show the transition fault coverage of the initial test pattern set T and of the enriched test pattern sets $T^E_{M\_1\_1}$, $T^E_{M\_1\_2}$, $T^E_{M\_2\_1}$ and $T^E_{M\_2\_2}$. The fault coverage is expressed in percent. The best transition fault coverages are shown in bold, and the worst results, which are below of fault coverages of initial test sets, are in italic.

Table 5. Results of transition fault simulation: complete functional delay SIT test

Circuit	T1	$T1^E_{M\_1\_1}$	$T1^E_{M\_1\_2}$	$T1^E_{M\_2\_1}$	$T1^E_{M\_2\_2}$
c432	95.57	**96.66**	96.58	96.44	96.58
c499	94.40	*87.35*	94.40	*87.35*	94.40
c880	98.91	99.33	**99.54**	99.33	**99.54**
c1355	97.13	*89.82*	97.13	*89.82*	97.13
c1908	95.24	*94.97*	**95.85**	*94.97*	**95.85**
c2670	96.51	**98.32**	**98.32**	**98.32**	**98.32**
c3540	83.08	**95.71**	88.15	95.68	88.02
c5315	98.41	**99.55**	99.54	**99.55**	99.53
c6288	99.75	99.75	**99.90**	99.75	**99.90**
c7552	97.05	**99.05**	98.82	99.05	98.83
Average	95.61	96.05	**96.82**	96.03	96.81

Table 6. Results of transition fault simulation: complete functional delay MIT test

Circuit	T2	$T2^E_{M\_1\_1}$	$T2^E_{M\_1\_2}$	$T2^E_{M\_2\_1}$	$T2^E_{M\_2\_2}$
c432	96.08	**97.82**	97.24	96.73	97.46
c499	93.00	*91.22*	**93.47**	*88.98*	93.18
c880	99.67	*99.29*	**99.83**	*99.46*	**99.83**
c1355	95.01	*92.18*	**95.64**	*90.69*	95.13
c1908	94.58	**95.21**	95.11	95.05	**95.21**
c2670	98.21	98.87	99.19	98.62	**99.29**
c3540	94.21	96.48	**94.69**	96.41	94.44
c5315	99.91	99.96	**99.98**	99.96	**99.98**
c6288	99.88	**99.89**	**99.89**	**99.89**	**99.89**
c7552	99.17	*99.15*	**99.45**	*99.17*	99.35
Average	96.97	97.01	**97.45**	*96.50*	97.38

Table 7. Results of transition fault simulation: incomplete functional delay MIT test

Circuit	T3	$T3^E_{M\_1\_1}$	$T3^E_{M\_1\_2}$	$T3^E_{M\_2\_1}$	$T3^E_{M\_2\_2}$
c432	87.28	87.65	87.28	88.44	87.28
c499	91.43	*90.61*	**92.42**	*88.37*	*90.06*
c880	90.90	98.46	**99.21**	98.58	98.71
c1355	92.36	*92.03*	**93.61**	*90.54*	*91.31*
c1908	81.58	82.03	82.18	83.15	**83.95**
c2670	90.44	91.11	92.38	**95.15**	94.95
c3540	88.28	91.84	91.60	**92.00**	91.74
c5315	97.94	**98.52**	98.32	98.43	*97.89*
c6288	98.72	98.72	98.72	98.72	98.72
c7552	97.84	*97.78*	**98.16**	97.91	98.07
Average	91.68	92.88	**93.39**	93.13	93.27

From Tables 5-7, it can be seen that the results of transition fault simulation are not so unambiguous as in case of the improvement of the average number of functional delay fault detections. The test enrichment accomplished using the proposed procedure EoFDT contributed to test quality improvement in eleven

cases out of twelve if we take into account average transition fault coverage. The average improvement of transition fault coverage ranges from 0.04% (Mode M_1_1, MIT tests T2) to 1.71% (Mode M_1_2, MIT tests T3). However the test enrichment Mode M_2_1 produces test sets which transition fault coverage is on average 0.47% lower than of initial test sets despite of the fact that the average number of functional delay fault detections was increased at 58.7% on average. Therefore, we can conclude that the increment of detections of targeted faults not always brings increment in detection of untargeted faults and defects. The reasons of such phenomenon are in changed conditions of signal propagation from circuit input to circuit output. This problem is investigated in detail in [31]. However, the statement that *n*-detection tests are effective in detecting untargeted faults and defects is true when the initial test sets stay unchanged, and additional tests patterns are generated in order to increase the number of detections of targeted faults. In our case, the procedure EoFDT modifies the patterns of the initial test set, and there is no test size increment.

If we examine tests for separate circuits, we can see that the test enrichment contributed to augmentation of fault coverage of untargeted faults in 86 cases out of 120, and in 22 cases out of 120 there was reduction of fault coverage of untargeted faults. These 22 cases belong to test set enrichment Modes M_1_1, M_2_1 and M_2_2. Contrarily, application of procedure EoFDT in Mode M_1_2 allowed us to improve the transition fault coverage in 26 cases out of 30 and there was no reduction of transition fault coverage. Therefore, if the primary goal of increasing of the number of functional delay fault detections is detecting of untargeted structural level faults and defects, we have to use the procedure EoFDT in Mode M_1_2.

At the end of this section we emphasise, however, that detection of transition faults is used only as an indication of the relative delay defect coverage achieved by various enriched functional delay fault test sets. All delay defect models are expected to benefit from the use of enriched in operation Mode M_1_2 functional delay fault test sets compared to conventional functional delay fault test sets. For example, for crosstalk faults, the detection of each functional delay fault more than one time increases the likelihood that one of the tests for the fault would create the appropriate conditions to activate the crosstalk fault and detect it.

5. Concluding remarks

We described an approach for functional test enrichment. The proposed postprocessing procedure modifies each test pattern of the test in such way that the modified test pattern detects all functional delay faults detectable on the initial test pattern and some additional functional delay faults. The test enrichment procedure does not increase the test size and it is fast because the procedure does not require test generation. The described approach enriches the test patterns using functional delay fault simulation. The performed experiments demonstrated that the proposed postprocessing procedure for functional delay test enrichment is an efficient and simple way to enhance the quality of initial test pattern set.

References

[1] **I. Pomeranz, S.M. Reddy.** On testing delay faults in macro-based combinational circuits. *Proceedings of International Conference on Computer-Aided Design, San Jose*, CA, 1994, 332-339.

[2] **F. Ferrandi, F. Fummi, G. Pravadelli, D. Sciuto.** Identification of design errors through functional testing. *IEEE Transactions On Reliability*, *Vol.*52, *No*.4, 2003, 400-412.

[3] **H. Kim, J.P. Hayes.** Realization-independent ATPG for designs with unimplemented blocks. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 20, *No*.2, 2001, 290–306.

[4] **F. Ferrandi, F. Fummi, D. Sciuto.** Implicit test generation for behavioral VHDL models. *Proceedings of International Test Conference,* 18-23 *October* 1998, 587-596.

[5] **E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas.** The use of software prototype for verification test generation. *Information Technology and Control*, *Vol.*37, *No*.4, 2008, 265–274.

[6] **V. Jusas, K. Motiejūnas.** Generation of functional delay test with multiple input transitions. *Information Technology and Control*. *Vol.*36, *No*.3, 2007, 259–267.

[7] **J. Zeng, M. Abadir, G. Vandling, L. Wang, A. Kolhatkar, J. Abraham.** On correlating structural tests for speed binning of high performance design. *Proceedings of the International Test Conference* ITC'04, 2004, 31-37.

[8] **T.N. Pham, F. Clugherty, G. Salem, J.M Crafts, J. Tetzloff, P. Moczygemba, T.M. Skergan.** *Functional test and speed/power sorting of the IBM POWER6 and Z10 processors. Proceedings of the International Test Conference* ITC'08, 2008, 1-7.

[9] **J. Yi, J.P. Hayes.** The coupling model for function and delay Faults. *Journal of Electronic Testing: Theory and Applications*, *Vol.*21, *No*.6, 2005, 631-649.

[10] **K.M. Butler, M.R. Mercer.** Assessing fault model and test quality. *Kluwer Academic*, 1992.

[11] **N. Neophytou, M.K. Michael, S. Tragoudas.** Functions for quality transition-fault tests and their applications in test-set enhancement. *IEEE Transactions on Computer Aided-Design of Integrated Circuits and Systems*, *Vol.*25, *No*.12, 2006, 3026-3035.

[12] **I. Pomeranz, S.M. Reddy.** Tuple detection for path delay faults: a method for improving test set quality. *Proceedings of the* 18*th International Conference on VLSI Design held jointly with* 4*th International Conference on Embedded Systems Design*, 2005, 41-46.

[13] **I. Pomeranz, S.M. Reddy.** Pattern sensitivity: a property to guide test generation for combinational circuits. *Proceedings of* 8*th Asian Test Symposium*, 1999, 75-80.

[14] E. Bareiša, V. Jusas, K. Motiejūnas, Š. Packevičius, R. Šeinauskas. The improvement of test independence from circuit realization. *Information technology and control*, *Vol.*33, 2004, 45-52.

[15] E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas. Functional test generation based on combined random and deterministic search methods. *Informatica*, 2007, *Vol.*18, *No*.1, 3-26.

[16] B. Benware, C. Schuermyer, N. Tamarapalli, K.-H. Tsai, S. Ranganathan, R. Madge, J. Rajski, P. Krishnamurthy. Impact of multiple-detect test patterns on product quality. *Proceedings of International Test Conference, September* 2003, 1031–1040.

[17] S. Venkataraman, S. Sivaraj, E. Amyeen, S. Lee, A. Ojha, R. Guo. An experimental study of *n*-detect scan ATPG patterns on a processor. *Proceedings of VLSI Test Symposium, April* 2004, 23–28.

[18] I. Pomeranz, S.M. Reddy. Worst-case and average-case analysis of *n*-detection test sets. *Proceedings of Design Automation Test European Conference, March* 2005, 444-449.

[19] S. Neophytou, M.K. Michael. On the relaxation of *n*-detect test sets. *Proceedings of the* 26*th IEEE VLSI Test Symposium*, 2008, 187-192.

[20] J. Geuzebroek, E.J. Marinissen, A. Majhi, A. Glowatz, F. Hapke. Embedded multi-detect ATPG and its effect on the detection of unmodeled defects. *Proceedings of the IEEE International Test Conference* ITC 2007, 2007, 1-10.

[21] I. Pomeranz, S.M. Reddy. Forming *n*-detection test sets without test generation. *ACM Transactions on Design Automation of Electronic Systems*, *Vol.*12, *No*.2, 2007, *Article No*.18.

[22] I. Pomeranz, S. M. Reddy. On *n*-detection test sets and variable *n*-detection test sets for transition Faults. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, *Vol.*19, *No*.3, 2000, 372-383.

[23] E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas. Properties of variable *n*-detection functional delay fault tests. *Information Technology and Control*, *Vol.*37, *No*.2, 2008, 95-100.

[24] I. Pomeranz, S. M. Reddy. On the saturation of *n*-detection test generation by different definitions with increased *n*. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, *Vol.*27, *No*.5, 2008, 946-957.

[25] E.Bareiša, V.Jusas, K.Motiejūnas, R.Šeinauskas. On the enrichment of static functional test. *Electronics and electrical engineering, Vol*.91, *No*.3, 2009, 9-14.

[26] 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, 54–163.

[27] E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas. Functional test transformation to improve compaction. *Electronics and Electrical Engineering, Kaunas, Technologija*, *Vol.*83, *No*.3, 2008, 53-58.

[28] E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas. The realization-independent testing based on the black box fault models. Iformatica, *Vol.*16, *No*.1, 2005, 19-36.

[29] E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas. Functional delay test construction approaches. *Electronics and Electrical Engineering*, *Vol.*74, *No*.2, 2007, 49-54.

[30] **K.M. Butler, M.R. Mercer.** Quantifying nontarget defect detection by target fault test sets. *Proceedings of the* 1991 *European Test Conference, Apr*. 1991, 91-100.

[31] **V.Jusas, K.Motiejūnas.** Impact of functional delay test compaction on transition fault coverage. *Information Technology and Control*, *Vol.*36, *No*.2, 2007, 196-201.

Received April 2009.

DOI: 10.5755/j01.itc.38.3.12124