ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 *2008. No. 3(83)*
ELEKTRONIKA IR ELEKTROTECHNIKA

ELECTRONICS
T170
ELEKTRONIKA

Functional Test Transformation to Improve Compaction

E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas
Software Engineering Department, Kaunas University of Technology,
Studentų str. 50-406, Kaunas, Lithuania, phone: +370 37 454229, e-mail: kestutis.motiejunas@ktu.lt

Introduction

The complexity of digital devices is growing continuously. The testing problem is becoming harder and harder. The construction of test patterns requires more time and is becoming the most crucial part of overall design process that delays the time-to-market of the digital device. Intellectual property (IP) components are used more widely in chip design; such 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 size of a functional test is usually much larger than that of an implementation-dependent one to assure good fault coverage for many implementations. When the synthesis of a high level description into a particular implementation is completed, the minimization of the functional test according to the particular implementation can be provided in order to exclude the test patterns that do not detect the faults of the particular implementation. Next, the list of undetected faults can be formed, and the deterministic methods can be used to detect the faults from this list. The adaptation of the functional test according to the particular implementation is much simpler than a generation of the test from the scratch. The process of adaptation doesn't require the long hours and it has a weak impact on the overall time of the design. That is a strong advantage of the functional test. If the high level description is resynthesized, the functional test remains the same. It has to be only adapted to the new implementation.

Compact test sets are very important for reducing the cost of testing the very large scale integrated (VLSI) circuits by reducing the test application time. This is especially important for the scan-based circuits as the test application time for these circuits is directly proportional to product of the test set size and the number of storage elements used in the scan chain [1].

In this work we present a new procedure of functional static-based test transformation into functional delay test that allows improving test compaction. We consider the combinational and fully enhanced-scanned sequential circuits. At the high level of abstraction every bit of the state variables is regarded as extra input and output, and consequently the whole of state variables could be regarded as a virtual scan register. Thus, the test patterns at the functional level are generated for the software prototype model, which has no state variables. When synthesis of the circuit is completed, the bits of the virtual scan register are linked to the flip-flops of the particular implementation.

The rest of the paper is organized as follows. Section 2 presents the related work. The proposed new

transformation procedure is presented in Section 3. The experimental results are given in Section 4. Finally, Section 5 presents the conclusions.

Related work

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 [2]. Under functional delay fault models proposed in [3-5], a fault is a tuple (I, O, t I, t O), where I is a CUT input, 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*n*m, where n is the number of inputs of the CUT and m is the number of outputs of the CUT. Under the model introduced in Underwood et al. [3], only one pair of test patterns must be generated per fault. This model was expanded in Pomeranz and Reddy [5] by considering Δ different test patterns per fault. Δ is a positive integer, usually in the low hundreds, and is given as an input parameter for each CUT. Pomeranz and Reddy [4] proposed that all possible patterns are generated for each fault. This model guarantees detection of all robustly testable path delay faults in any gate-level implementation. However, the resulting test set sizes, as well as the test generation times, are very large and make this model impractical, especially for large circuits [4, 5]. The studies in [5] showed that it is not necessary to generate all possible test patterns for each fault in order to guarantee that actual path delays are covered in some gate-level implementation of the function.

Another fault model for functional ATPG based on input-output stuck-at faults testing and called pin pair (PP) fault model is suggested by Bareiša et al. in [6] and generalized in [7].

Now we provide a brief presentation of the main concepts of this model. The behavioral view or the "black-box" represents the system by defining the behavior of its outputs according to the values applied on its inputs without the knowledge of its internal organization. In this case, the input-output relationship can be determined only. Let the circuit have a set of inputs $X = \{x_1, x_2, ..., xi, ... , x_n\}$ and a set of outputs $Z = \{z_1, z_2, ... ,z_j, ... ,z_m\}$. The pin fault model considers the stuck-at-0/1 faults occurring at the module 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 2*n+2*m possible pin faults. Input-output pin 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*n*m. Note that in general it is not possible to relate the PP fault with the defects of the module unambiguously, because the PP fault doesn't fix exactly the signal propagation path in the circuit.

The PP fault test transformation into functional delay fault test rule was presented in [8]. We recall this rule.

Rule. If the input pattern *q* detects the PP fault (x_i^t, z_j^k), t=0,1, k=0,1, then the pair of input patterns <*p*, *q*>, where the signal value of input x_i in the pattern *q* is $\bar{t}$ and the signal value of input x_i in the pattern *p* is t, detects the functional delay faults: (x_i, z_j, r x_i, r z_j), when t=0, k=0; (x_i, z_j, r x_i, f z_j), when t=0, k=1; (x_i, z_j, f x_i, r z_j), when t=1, k=0; (x_i, z_j, f x_i, f z_j), when t=1, k=1.

For example, consider the circuit represented as black-box in Fig. 1. Let's say, that circuit response to input pattern <10011> is <01> and that this pattern detects the PP faults $(X2^1,Z1^1)$, $(X2^1,Z2^0)$, $(X3^1,Z1^1)$ and $(X5^0,Z1^1)$.

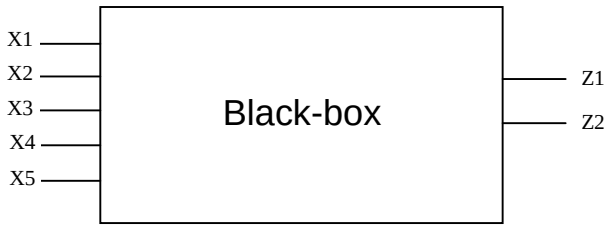

Fig. 1. Example circuit

The transformation of the pattern <10011> according to *Rule* results into the sequence of input patterns pairs (<11011,10011>,<10111,10011>,<10010,10011>). This sequence detects the functional delay faults (X2, Z1, f X2, f Z1), (X2, Z2, f X2, r Z1), (X3, Z1, f X3, f Z1) and (X5, Z1, r X5, f Z1).

Note that every of PP tests according to *Rule* composed test pair is single-input transition test (SIT) [2] and, therefore, every test pair propagates the transition from a primary input to a primary output in a function robust manner [2]. Another observation is that the test generation for PP faults can be accomplished using various approaches: 1. One test is generated for each PP fault in the circuit; 2. All possible tests are generated for each PP fault in the circuit; 3. Δ tests are generated for each PP fault in the circuit. Thus the functional delay fault tests obtained from PP tests generated using approaches 1, 2 or 3 correspond to tests generated using models described in [3], [4] and [5], respectively. In this paper we consider the case when one test is generated for each PP fault in the circuit.

Pin pair test transformation into functional delay test procedure

The test generation problem is compounded by the fact that traditional gate-level ATPG techniques are finding it difficult to handle the size and complexities of the current generation of circuits. However, the top-down design philosophy offers an alternative approach to ease test generation. It offers a high-level view of the circuit (e.g., algorithmic circuit description or register-transfer level (RTL)) that has a significantly lower number of primitives than at the gate level. Consequently, test generation can be performed at high level of abstraction making the problem more tractable. While high–level test generation techniques enable fast test generation, they do not systematically address the process of finding compact test sets [1].

The test pattern pairs constructed according *Rule* posses the change of signal value only on single input. Suppose we have an input pattern *w* that detects *q* PP faults. Thus for detection of *q* corresponding functional delay faults, maximum *k*<= *q* pairs of input patterns are built on the base of this pattern (signal transition on one input can cause signal transitions on *s* outputs, consequently, only one pair of input patterns is needed for detection of *s* functional delay faults). Let's continue the analysis of our example. The considered input pattern <10011> detects 4 PP faults, though for detection corresponding functional delay faults only 3 pattern pairs are built because the signal transition on input X2 invokes the signal transitions on both outputs Z1 and Z2. Therefore, the number of obtained using *Rule* test pattern pairs is equal to the number inputs related with PP fault detection on considered test pattern. In our case, we have 3 such inputs and, consequently, 3 test pattern pairs are built.

There is another way described in [9] to obtain functional delay fault tests from PP tests. By applying the approach from [9] every input pattern that detects PP faults is transformed only into one input pattern pair in such a way that the signal value transition occurs on every input that is related with PP fault detection on the considered test pattern. For our example, input pattern pair <11110,10011> is constructed. Consequently, if the test for PP faults consists of *p* input patterns the constructed functional delay fault test has *p* input pattern pairs too. Thus the obtained test is much shorter than by applying *Rule*. The experiments on ISCAS'85 benchmark circuits demonstrated the test shortening of 3.8 times on average [9]. However, the test pattern pairs constructed by applying the approach from [9] possess the change of signal value on more than one input. Therefore, these pattern pairs are multi-input transition (MIT) tests [2] and some of functional delay faults that are functional robustly detectable on SIT test may become functional nonrobustly detectable [2] or even worse not detectable on considered test pattern pair, because some activation conditions needed for signal transition propagation from particular input to particular output may be corrupted. We made an extra experiment on benchmark circuit c432. The functional delay fault test constructed according to *Rule* contained 348 test pattern pairs and covered 100% of functional delay faults, whereas the applying of approach from [9] produced 117 test pattern pairs and this test covered only 81% of functional delay faults.

Now our goal is to propose compromise between these two discussed approaches, i.e. to compose a procedure that enables pin pair test transformation into compact functional delay fault test without loss of fault coverage. The only way to compact functional delay fault tests is in enabling multi-input transitions in test pattern pairs. Therefore, we have to define the necessary conditions for the transition propagation from the considered input to considered output by more than one signal change in the test pattern pair.

Let's say:

1. There is the multi-input transition test pattern pair <*p*, *q*>;
2. The signal value on input x_i in the pattern *p* is t and the signal value on input x_i in the pattern *q* is $\bar{t}$;
3. The signal value on output z_j as circuit response to input pattern *p* is k and the signal value on output z_j as circuit response to input pattern *q* is $\bar{k}$.

Lemma. If after signal value fixation to t on single input x_i in both test patterns *p* and *q* the signal value on output z_j as circuit response to input patterns *p* and *q* is stable k, than this circuit response is caused only by signal value fixation on the input x_i.

Proof. Suppose the stable signal value k on output z_j as circuit response to input patterns *p* and *q* is caused by signal values on the input x_r. Thus the different circuit response on output z_j could evoke solely the signal value change on the input x_r either in both test patterns *p* and *q* or in one test pattern *p* or *q*. However it is a contradiction to Lemma statement that the signal value change occurs only on one input x_i. Therefore, the stable signal value k on output z_j as circuit response to input patterns *p* and *q* is caused by signal value fixation to t on one input x_i.

Corollary 1. The multi-input transition test pattern pair <*p*, *q*> detects the functional delay fault (x_i, z_j, $t\rightarrow\bar{t}$ x_i, $k\rightarrow\bar{k}$ z_j).

It is desirable that the generated multi-input transition test would guarantee the function-robust propagation (FRP) property. According to definition from [2] a transition *tI* on input *I* is function- robustly propagated as a *tO* transition to output *O* when the signal value on *O* changes *if and only if* the signal value on *I* changes. Now we can formulate the next corollary.

Corollary 2. The signal value transition $t\rightarrow\bar{t}$ on input x_i is function- robustly propagated as the signal value transition $k\rightarrow\bar{k}$ to output z_j.

At this point on basis of Lemma and Corollary 1 we present a pseudocode description of procedure of pin pair test transformation into compact functional delay test (PPTTiCFDT) (see Fig. 2).

The meaning of essential variables used in the PPTTiCFDT procedure: X- matrix for detectability marking of functional delay faults; n – number of inputs; m – number of outputs; t – number of composed test pattern pairs; L – PP fault test size; P – test pattern. All other variables are local variables used for calculation organization. Expression Z:=f(P) implies circuit simulation, i.e. Z is circuit response to input pattern P. The lines 4-10 are used for marking of circuit inputs related with PP fault detection (denominate them as active inputs) in test pattern P_k and for building of test pattern pair <P^1, P^2> where P^2:=P_k while the signal values of active inputs of pattern P^1 are inverted in regard to signal values available in test pattern P_k. The other part of pseudocode (lines 11-34) describes the MIT test pattern pairs, which satisfy FRP property, formation on the basis of <P^1, P^2>.

```
1.  PROCEDURE PPTTiCFDT();
2.    X=||x_{i,j}:=0||_{2n x 2m}; t:=1;
3.    DO P_1,P_2, ....,P_k,...,P_L;
4.        PZW:=0; H=||h_i:=0||_n; P^2:=P_k; P^1:=P_k; // P^2=(p^2_1,p^2_2,...,p^2_i,...,p^2_n), p^2_i∈ (0,1)
5.        Z^2:=f(P^2); // Z^2= (z^2_1,z^2_2,...,z^2_j,...,z^2_m), z^2_j∈ (0,1)
6.            DO i:=1,2,3,…,n; p^2_i:=NOT(p^2_i); d:= p^2_i; Z‘=f(P^2);
7.                DO j:=1,2,3,…,m; c=1- z^2_j;
8.                    IF (z^2_j≠ z‘_j) AND (x_{2i-d,2j-c} =0) THEN h_i:=1; p^1_i:= p^2_i; PZW:=1; ENDIF;
9.                ENDDO; p^2_i:=NOT(p^2_i);
10.           ENDDO;
11.           DO WHILE (PZW=1); PZW:=0; P’:=P^1; Z^1:=f(P^1); B:=0; WRITE:=0; REMAIN:=0;
12.               DO i:=1,2,3,…,n; PZ:=0;
13.                   IF (h_j=1) THEN
14.                       p^2_i:= NOT(p^2_i); d:=p^2_i; Z‘=f(P^2);
15.                       DO j:=1,2,3,…,m; c=1- z^2_j;
16.                           IF (z^2_j≠ z‘_j) THEN
17.                               IF (z^1_j≠ z‘_j) THEN
18.                                   IF x_{2i-d,2j-c} =0 THEN PZ:=1; ENDIF;
19.                               ELSE
20.                                   IF x_{2i-d,2j-c} =0 THEN x_{2i-d,2j-c}:=1; ENDIF;
21.                               ENDIF;
22.                           ENDIF;
23.                       ENDDO; p^2_i:=1- p^2_i;
24.                       IF (PZ=1) THEN REMAIN:=1; B:=i;
25.                       ELSE p‘_i:= p^2_i; h_i:=0; WRITE:=1; ENDIF;
26.                   ENDIF;
27.               ENDDO;
28.               IF (WRITE=1) THEN PZW:=1; P^t:=P^1; P^{t+1}:=P^2; t:=t+2; P^1:=P‘;
29.               ELSE
30.                 IF (REMAIN=1) THEN
31.                     PZW:=1; P^D:=P^2;p^D_B:=1- p^D_B; h_B:=0; P^t:=P^D; P^{t+1}:=P^2; t:=t+2;  p^1_B:=1- p^1_B;
32.                 ENDIF;
33.               ENDIF;
34.           ENDWHILE;
35.   ENDDO;
36. END PROCEDURE PPTTiCFDT;
```

Fig. 2. Pseudocode for the PPTTiCFDT procedure

Experimental results

The non-redundant ISCAS’85 and ITC’99 benchmark circuits have been selected for experiments. The test sets for PP faults were generated for the black-box model of the circuits [7] using a random search procedure. Remind the black-box model represents a system by defining the behavior 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 programming language C were used by the test generation for the PP faults.

In this work, we presented a new functional pin pair fault test transformation into functional delay test procedure that allows improving test compaction. There are many publications [2-5, 8-10] devoted to functional delay faults testing. However the obtained test sizes are presented only in [8-10]. Therefore we will compare our experimental results with results derived using approaches from [8] (in [8] and [9] the same approach is used) and [10]. Table 1 and Figure 3 give results for pin pair fault test transformation into functional delay test using PPTTiCFDT procedure and approaches [8] and [10]. Note that all presented in Table 1 tests posses 100% coverage of targeted faults, i.e. functional delay faults, and that one test is generated for each targeted fault in the circuit. In [10] there is no information for ITC’99 benchmark circuits and for the circuit c1355, whereas the fault coverage of functional delay faults for the circuit c6288 is less than 100% (90.3%). The test size reduction expressed as two test sizes ratio is presented in columns under heading “TSR”.

If we examine the results presented in Table 1 and Fig. 3 we can see that in all cases the proposed PPTTiCFDT procedure produced superior results. By comparing approach [8] and PPTTiCFDT the test size reduction is on average 1.78 and ranges from 1.18 (circuit c6288) to 4.83 (circuit c1355). The comparison of approach [10] and PPTTiCFDT shows even better results: the test size reduction spans between 1.74 (circuit c1908) and 3.43 (circuit c2670) and is on average 2.87. Remember that approaches [8] and [10] generate single-input transition tests whereas the PPTTiCFDT procedure creates multi-input transition tests, however all considered tests propagate the transitions from input to output in functional robust manner, i.e. all tests satisfy the FRP property.

Table 1. Test sizes and test size reductions

Circuit	Test size		TSR	Test size		TSR	Test size		TSR
	Approach [8]	PPTTiCFDT		Approach [10]	Approach [8]		Approach [10]	PPTTiCFDT	
c432	348	244	1.43	807	348	2.32	807	244	3.31
c499	5180	1159	4.47	3917	5180	0.76	3917	1159	3.38
c880	1001	743	1.35	1325	1001	1.32	1325	743	1.78
c1355	5162	1068	4.83	-	-	-	-	-	-
c1908	2359	1814	1.30	3159	2359	1.34	3159	1814	1.74
c2670	1820	1153	1.58	3958	1820	2.17	3958	1153	3.43
c3540	1457	1166	1.25	2510	1457	1.72	2510	1166	2.15
c5315	4950	3382	1.46	10675	4950	2.16	10675	3382	3.16
c6288	1065	903	1.18	-	-	-	-	-	-
c7552	5801	4331	1.34	13820	5801	2.38	13820	4331	3.19
b04	1364	1080	1.26	-	-	-	-	-	-
b08	290	230	1.26	-	-	-	-	-	-
Average	2914	1596	1.78	5021	2865	1.75	5021	1749	2.87

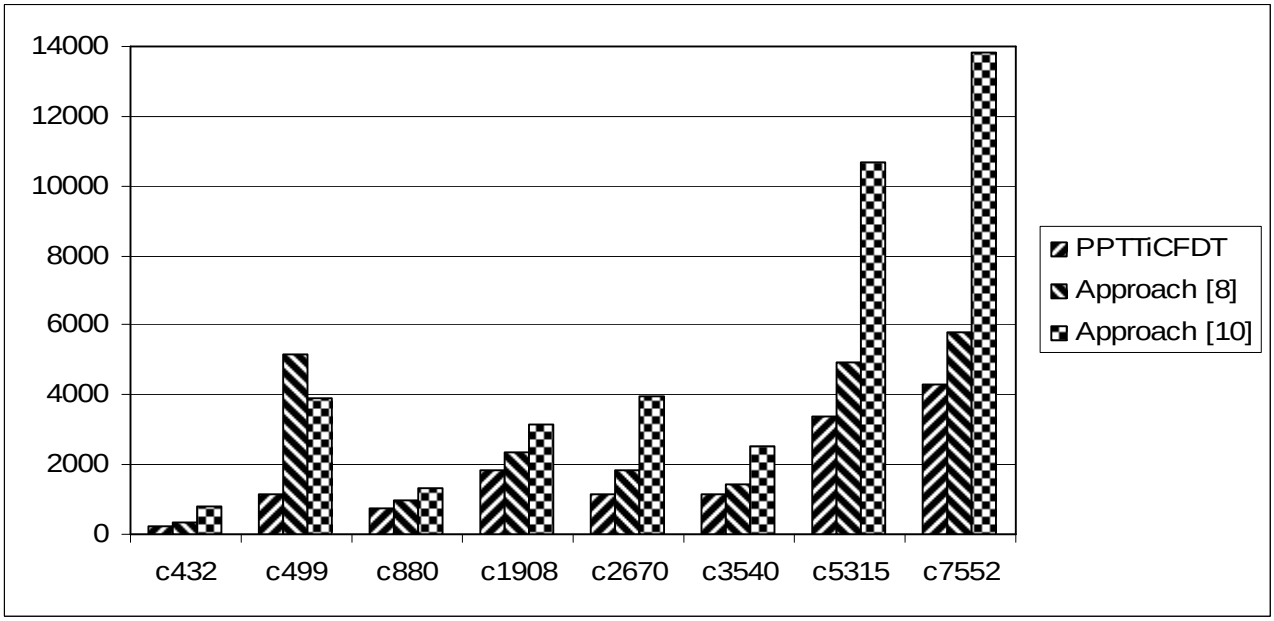

Fig. 3. Test size comparison

The comparison of approaches [8] and [10] is interesting too because both of them produce SIT tests. Our technique more widely described in [8] provides with shorter tests in seven of eight cases and the test size reduction is on average 1.75. There is only one exception, namely the circuit c499 for witch the approach [10] produced 1.32 times shorter test set.

Conclusions

In this paper, we defined the necessary conditions for the transition propagation from the considered input to considered output in the multi-input transition test pattern pair. We proved that the composed according these conditions test pattern pairs guaranty function-robustly propagation of signal value transition on the input to the output. On basis of proved lemma and its corollaries we developed a functional pin pair fault test transformation into functional delay test procedure that allowed improving test compaction. Experimental results for ISCAS'85 and ITC'99 benchmark circuits showed that the proposed approach reduced test size up to 4.83 times comparing with results presented in other publications whereas average test size reduction was up to 2.87 times.

References

1. **Ravi S., Lakshminarayana G., Jha N. K.** High-Level Test Compaction Techniques // IEEE Transactions on Computer Aided Design of Integrated Circuit and Systems. – 2002. – Vol. 21, No. 7. – P. 827–841.
2. **Michael M., and Tragoudas S.** ATPG Tools for Delay Faults at the Functional Level // ACM Transactions on Design Automation of Electronic Systems. – January 2002. – Vol. 7, No. 1. – P. 33–57.
3. **Underwood B., Law W. O., Kang S., Konuk H.** Fastpath: A path-delay test generator for standard scan designs //

Proceedings of 1994 International Test Conference. – 1994. – P. 154–163.
4. **Pomeranz I., and Reddy S. M.** On testing delay faults in macro-based combinational circuits // Proceedings of 1994 International Conference on Computer-Aided-Design. – 1994. – P. 332–339.
5. **Pomeranz I., and Reddy S. M.** Functional test generation for delay faults in combinational circuits // Proceedings of 1995 International Conference on Computer-Aided-Design. – 1995. – P. 687–694.
6. **Bareisa E., Seinauskas R.** Test Selection Based on the Evaluation of Input Stuck-at Faults Transmissions to Output // Information technology and control. – Kaunas: Technologija, 1996. – No. 2(3). – P. 15–18.
7. **Bareiša E., Jusas V., Motiejūnas K., Šeinauskas R.** The Realization-Independent Testing Based on the Black Box Fault Models // Informatica, 2005. – Vol. 16, No. 1. – P. 19–36.
8. **Bareiša E., Jusas V., Motiejūnas K., Šeinauskas R.** Application of Functional Delay Tests for Testing of Transition Faults and Vice Versa // Information Technology And Control. – Kaunas: Technologija, 2005. – Vol. 34, No. 2. – P. 95–101.
9. **Bareiša E., Jusas V., Motiejūnas K., Šeinauskas R.** Functional Delay Test Construction Approaches // Electronics and Electrical Engineering. – Kaunas: Technologija, 2007. – No. 2(74). – P. 49–54.
10. **Michael M., and Tragoudas S.** ATPG Tools for Delay Faults at the Functional Level // Proceedings of Design. Automation and Test in Europe Conference and Exhibition. – 1999. – P. 631–635.

Submitted for publication 2007 03 15

E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas. Functional Test Transformation to Improve Compaction // Electronics and Electrical Engineering. – Kaunas: Technologija, 2008. – No. 3(83). – P. 53–58.

Compact test sets are very important for reducing the cost of testing the very large scale integrated circuits by reducing the test application time. In this work we presented a new procedure of functional static-based test transformation into functional delay test that allows improving test compaction. Experimental results for ISCAS'85 and ITC'99 benchmark circuits showed that the proposed approach reduced test size up to 4.83 times comparing with results presented in other publications whereas average test size reduction was up to 2.87 times. Ill. 3, bibl. 10 (in English; summaries in English, Russian and Lithuanian).

Э. Барейша, В. Юсас, К. Мотеюнас, Р. Шейнаускас. Трансформирование функциональных тестов с целью уменьшения их объема // Электроника и электротехника. – Каунас: Технология, 2008. – № 3(83). – С. 53–58.

Короткие тесты очень важны для снижения цены тестирования очень больших интегральных схем. В данной работе предложен новый алгоритм трансформирования функциональных статических тестов в тесты задержки, позволяющий значительно сократить объем получаемого теста. С ISCAS'85 и ITC'99 эталонными схемами проведенные эксперименты показали, что в работе предложенный метод по сравнению с результатами, приведенными в других публикациях, сократил тест для отдельных схем до 4,83 раза, а среднее сокращение теста было до 2,87 раза. Ил. 3, библ. 10 (на английском языке; рефераты на английском, русском и литовском яз.).

E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas. Funkcinių testų transformavimas jų ilgiui sumažinti // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2008. – Nr. 3(83). – P. 53–58.

Trumpi testai yra itin svarbūs mažinant labai didelių integrinių schemų testavimo kainą. Pateiktame darbe pasiūlytas naujas algoritmas funkciniams statiniams testams transformuoti į vėlinimo gedimams tikrinti skirtus testus, leidžiantis gerokai sumažinti gaunamo testo apimtį. Su ISCAS'85 ir ITC'99 etaloninėmis schemomis atlikti eksperimentai parodė, kad darbe pasiūlytas metodas, palyginti su kitose publikacijose paskelbtais rezultatais, atskirų schemų testą sutrumpino iki 4,83 karto, o testų sutrumpinimo vidurkis buvo iki 2,87 karto. Il. 3, bibl. 10 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).