ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2005, Vol. 34, No. 2

APPLICATION OF FUNCTIONAL DELAY TESTS FOR TESTING OF TRANSITION FAULTS AND VICE VERSA

Eduardas Bareiša, Vacius Jusas, Kęstutis Motiejūnas, Rimantas Šeinauskas

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

Abstract. Rapid advances of semiconductor technology lead to higher circuit integration as well as higher operating frequencies. The statistical variations of the parameters during the manufacturing process as well as physical defects in integrated circuits can sometimes degrade circuit performance without altering its logic functionality. These faults are called delay faults. In this paper we consider the quality of the tests generated for two types of delay faults, namely, functional delay and transition faults. We compared the test quality of functional delay tests in regard to transition faults and vice versa. We have performed various comprehensive experiments with combinational benchmark circuits. The experiments exhibit that the test sets, which are generated according to the functional delay fault model, obtain high fault coverages of transition faults. However, the functional delay fault coverages of the test sets targeted for the transition faults are low. It is very likely that the test vectors based on the functional delay fault model can cover other kinds of the faults. Another advantage of test set generated at the functional level is that it is independent of and effective for any implementation and, therefore, can be generated at early stages of the design process.

1. Indroduction

Rapid advances of semiconductor technology lead to higher circuit integration as well as higher operating frequencies. Conventional fault models like the standard single stuck-at model were developed for gate-level logic circuits. Regardless of stuck-at fault model's efficiency for several decades, alternative models need to account for deep sub-micron manufacturing process variations [1]. Increasing performance requirements of circuits make it difficult to design them with large timing margins. Thus imprecise delay modelling, statistical variations of the parameters during the manufacturing process as well as physical defects in integrated circuits can sometimes degrade circuit performance without altering its logic functionality. These faults are called delay faults. Ensuring that the designs meet the performance specifications requires application of delay tests. However, delay fault testing of deep submicron designs is a complex task. It requires application of two-vector patterns at the circuit's intended operating speed. The test application usually requires high-speed testers or it could also be done through built-in self-test [2].

Two general types of delay fault models, the gate delay fault model [3, 4] and the path delay fault model [5], have been used for modelling delay defects. Although the path delay fault model is generally considered to be more realistic and effective in modelling physical delay faults, it is often difficult to use in practice due to a huge number of paths in the circuit. Therefore, the gate delay fault model is more feasible for large circuits. The most commonly used gate delay fault model is the transition fault model [3]. According to this model, every line in the circuit is associated with two transition faults: a slow-to-rise fault (rising fault) and a slow-to-fall fault (falling fault). To simplify the analysis of transition faults, it is often assumed that the extra delay caused by a transition fault on a line is sufficiently large such that the delay of every path passing through this line exceeds the maximum allowed value, which is usually the system clock period for synchronous sequential circuits. Because of this assumption, it is generally believed that tests results based on gate delay model are only useful for capturing large-size delay defects. To capture small-size distributed along specific circuit path defects, a path delay fault model is often used. However, the following problems are associated with the path delay fault model: the number of paths which are targeted by the test generation is very large; many path delay faults are not testable.

Test sets for path delay faults in circuits with large numbers of paths are typically generated for path delay faults associated with the longest circuit paths. This may lead to undetected failures since a shorter path may fail without failing of any of the longest paths. In the paper [6] I. Pomeranz and S. M. Reddy propose a test enrichment procedure that significantly increases the number of faults associated with the

next-to longest paths that are detected by a compact test set. An alternative approach to this problem is an optimisation of the critical path selection [7] or a selection of the longest testable path [8, 9]. In the papers [8, 9] the authors combine the merits of both the transition fault model and the critical path delay model. The authors of both papers agree that more automatic test pattern generation efforts are required to produce tests for all faults in this model than that given by the single transition fault model. Therefore, in the paper [8] it is suggested that in order to obtain a high quality transition fault test set using reasonable run times, initially a conventional transition fault test set can be generated and then augmented by a test based on the longest testable path passing through the fault site. One of limitations of the combined approach is that in case of certain distributed delay defects the derived tests will fail to detect some of the delay faults that are not targeted. There is considered only one path through any given line. However, there may be some other path of the same length or even shorter through the target line, with distributed delay defects exceeding the permissible propagation delay [10].

We have briefly surveyed various fault models that are applicable for gate-level circuit description and their advantages and limitations. An efficient fault model that will result in a high fault coverage and low computational complexity still remains to be elusive. At this time, there is no agreement on using a fault model, however, it is generally accepted that the path delay model is most comprehensive [10].

In the case 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, functional-level test generation must be performed. A test set generated at the functional level is independent of and effective for any implementation and, therefore, can be generated at early stages of the design process [11, 12]. Functional Automatic Test Pattern Generation (ATPG) can also be used to identify testability problems before an implementation is selected. Another advantage of functional ATPG for path delay faults over structural ATPG is related to the number of targeted faults. For structural ATPG, the number of faults is proportional to the number of paths in the circuit, which very often is 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 [12].

Functional delay fault models are proposed in [13-15]. The Underwood et al. [13] 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 [14] 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 [14] model is that it results in test sets of very large size. A compromise is mentioned in Pomeranz and Reddy [15] that results in fewer tests at the cost of reduced fault coverage. The functional fault model proposed here encompasses the Underwood et al. [13] and Pomeranz and Reddy [14] models in an attempt to combine their advantages.

In this paper we will analyse the situation when tests are generated for functional delay faults and applied for detection of transition faults and vice versa. The paper is organized as follows. We review the functional delay fault models in Section 2. We analyse the relationship between functional delay and pin pair fault models in Section 3. We explore functional delay tests vs. transition tests in Section 4. We finish with conclusions in Section 5.

2. Related work. Functional delay fault models

All the definitions in this section are taken from [12-15].

As mentioned in the introduction Underwood et al. [13] and Pomeranz and Reddy [14, 15] presented fault models for functional ATPG. Under these models, a fault is a tuple (I, O, tI, tO), where I is a CUT input, O is a CUT output, tI is a rising or falling transition at I , and tO is a rising or falling transition at O. Under the model introduced in Underwood et al. [13], only one pair of test patterns must be generated per fault. This model was expanded in Pomeranz and Reddy [15] 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 [14] 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 [14, 15]. However, the studies in [15] 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. The validity of the model in Pomeranz and Reddy [15] is verified by applying the generated test sets to various gate-level implementations [12, 15].

Definition 1. A functional delay fault is a tuple (I, O, tI, tO), where I is a CUT input, O is a CUT output, tI is a rising or falling transition at I, and tO is a rising or falling transition at O [13,14].

Thus, four functional delay faults are associated 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.

Definition 2. A test for the functional delay fault is a pair of input patterns $\langle u, v \rangle$ that propagates a transition from a primary input to a primary output of a circuit in a function-robust manner [12].

The function-robust propagation, referred to as the FRP property, is defined as follows:

Definition 3. A transition propagates function-robustly from a primary input I to a primary output O if the value on O does not change unless the value on I changes, independently of the order or speed at which the values of the other primary inputs change [12].

The above definition applies to fully specified pairs of patterns, i.e., all the values on all the primary inputs are known.

Definition 4. A single-input transition (SIT) test is a pair of input patterns <*u, v*> in which exactly one input assumes a transition [12].

Definition 5. A multi-input transition (MIT) test is a pair of input patterns <*u, v*> in which more than one input assume a transition [12].

Any SIT test always satisfies the FRP property. In the case of MIT test generation, an ATPG tool must explicitly determine that a generated test satisfies the FRP property. Thus, it must verify that the transition tO at the output O is caused only by the transition tI at the input I.

The three fault models are defined in [13-15] as follows:

Model M1 (as proposed by Underwood et al. [13]). One SIT or MIT test must be generated for each functional delay fault in the circuit.

Model M2 (as proposed by Pomeranz and Reddy [14]). All possible tests must be generated for each functional delay fault in the circuit.

Model M3 (as proposed by Pomeranz and Reddy [15]). Δ SIT or MIT tests must be generated for each functional delay fault in the circuit.

Δ is a parameter that can be adjusted according to the circuit size. Pomeranz and Reddy [15] discussed how to select an appropriate value for Δ. When Δ=1, the model M3 reduces to the model M1. When Δ is unlimited, the model M3 reduces to the model M2.

3. The relationship between functional delay and pin pair fault models

Another model for functional ATPG based on input-output paths testing and called pin pair fault model is suggested by Bareiša et al. in [16] and generalized in [17].

Now we provide a brief presentation of the main concepts of this model. Let the circuit have a set of inputs $X = \{x_1, x_2, \ldots, x_i, \ldots, x_n\}$ and a set of outputs $Z = \{z_1, z_2, \ldots, z_j, \ldots, z_m\}$. 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 2n +2m 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. We denote the set of the pin pair faults by

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

The test vector detects the pin pair fault (x_i^t, z_j^k) of the module if the test vector detects both the pin faults x_i^t, and z_j^k of the pair on the output z_j of the module. It may appear that there exist no electric connections between the input and the output, and the pin pair fault defined by these inputs and outputs can't be detected. These faults are not testable. The PP fault (x_i^t, z_j^k) of a module is testable if a conventional deterministic test generator for a realization of the module finds a test vector, which detects a pin fault x_i^t on an output z_j while the input x_i and the output z_j are set up to the $\bar{t}$ and $\bar{k}$. The number of testable PP faults equals 4*n*m minus the number of not testable PP faults. 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.

For example, consider the circuit provided in Figure 1. The set of the testable PP faults of the circuit includes the faults P1 = $\{(a^1,y^1), (a^0,y^0), (d^0,y^0), (d^1,y^1), (b^1,y^1), (b^1,y^0), (b^0,y^1), (b^0,y^0), (c^1,y^1), (c^1,y^0), (c^0,y^1), (c^0,y^0)\}$. The six test vectors 1010, 1110, 0011, 0111, 1100, and 0101 detect all the PP faults. The test vector 1010 detects the PP faults (b^1,y^0), (a^0,y^0), the test vector 1110 detects the PP faults (b^0,y^1), (c^0,y^1), the test vector 0011 detects the PP fault (b^1,y^1) and so on.

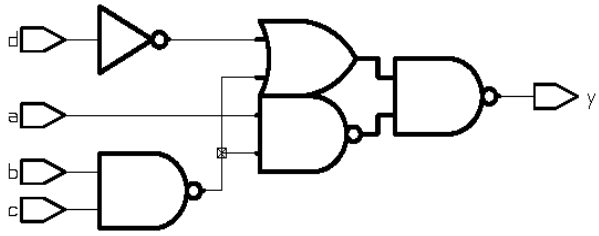

Figure 1. Example circuit

If we compare the two provided functional fault models, namely, functional delay and pin pair model we see that both models have almost the same meaning with one distinction: the functional delay model is intended for detection of malfunctions in the dynamic behaviour of CUT and the pin pair model – for detection of malfunctions in the steady state of CUT. For example, consider the PP fault (b^1,y^0) and functional delay fault (a, y, ra, fy) (see Figure 1). The PP fault (b^1,y^0) is detected by an input pattern 1010 and functional delay fault (b, y, rb, fy)is detected by a pair of input patterns <1010, 1110>.

Thus we can simply define the rules how to get a PP fault test from functional delay fault test and vice versa.

Rule 1. If the pair of input patterns <*u, v*> detects the functional delay fault (x_i, z_j, tr x_i, tr z_j), tr=r,f (r –

rising transition, f –falling transition), then the single input pattern *v* detects the PP fault (x_i^t, z_j^k), t=0,1, k=0,1.

Rule 2. 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 <*q, p*>, where the signal value of input x_i in the pattern *q* is 1 (0) and the signal value of input x_i in the pattern *p* is 0 (1) detects the functional delay fault (x_i, z_j, f(r) x_i, tr z_j), tr=r,f.

Suppose we have an input pattern *w* that detects *l* PP faults. Then it is needed for the transformation of this pattern for detection of *l* corresponding functional delay faults to build maximum *l* pairs of input patterns according to Rule2 (signal transition on one input can cause signal transitions on *s* outputs, therefore, only one pair of input patterns is needed for detection of *s* functional delay faults). For example, consider input pattern 1010 that detects the PP faults (b^1,y^0), (a^0,y^0). Transformation of this pattern results into the sequence of input patterns (<1010, 1110>, <1010, 0010>). This sequence detects functional delay faults (b, y, rb, fy) and (a, y, fa, fy).

Note that all of PP tests according to Rule 2 composed test pairs are single-input transition tests (see Definition 4) and, therefore, all test pairs satisfy the FPR property (see Definition 3). Another observation is that the test generation for PP faults can be accomplished using various modes: 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 tests obtained from PP tests generated using modes 1, 2 or 3 correspond to tests generated using Model M1, Model M2 and Model M3, respectively.

4. Functional delay tests vs. transition tests

An interesting issue is how the tests generated for one type of faults cover the faults of another type. In this paper we are going to compare the test quality of functional delay tests in regard to transition faults and vice versa. Both types of faults are designed for dynamic testing, however the test generation methods for these faults are different. The non-redundant ISCAS'85 benchmark circuits [18] have been selected for experiments. The functional delay tests have been got from PP fault tests according to Rule 2. The test sets for PP faults were generated for the black-box model of the circuits [17] using a random search procedure and Model M1, i.e. one test was generated for each PP fault in the circuit. 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 programming language C for ISCAS'85 benchmark circuits were used by the test generation for the PP faults. The Synopsys test pattern generator TetraMAX was used for test generation of transition faults.

The parameters of the non-redundant ISCAS'85 benchmark circuits are given in Table 1 and Figure 2. The connectivity rate demonstrates the relation between the number of testable functional delay (PP) faults and the total number of possible functional delay (PP) faults and is computed as follows:

Connectivity rate = the number of testable functional delay (PP) faults/ 4*n*m.

Table 1. Parameters of the non-redundant ISCAS'85 benchmark circuits

Circuit	Gates	Inputs n	Outputs m	4*n*m	Testable functional delay (PP) faults	Connectivity rate %	Transition faults
C432	160	36	7	1008	540	54%	1412
C499	202	41	32	5248	5184	99%	3430
C880	383	60	26	6240	1326	21%	2396
C1355	546	41	32	5248	5184	99%	3350
C1908	880	33	25	3300	3004	91%	4848
C2670	1193	157	64	40192	3320	8%	5646
C3540	1669	50	22	4400	2588	59%	8960
C5315	2307	178	123	87576	10540	12%	13816
C6288	2406	32	32	4096	3068	75%	14422
C7552	3512	206	107	88168	12188	14%	19160
Total	13258			245476	46942		77440

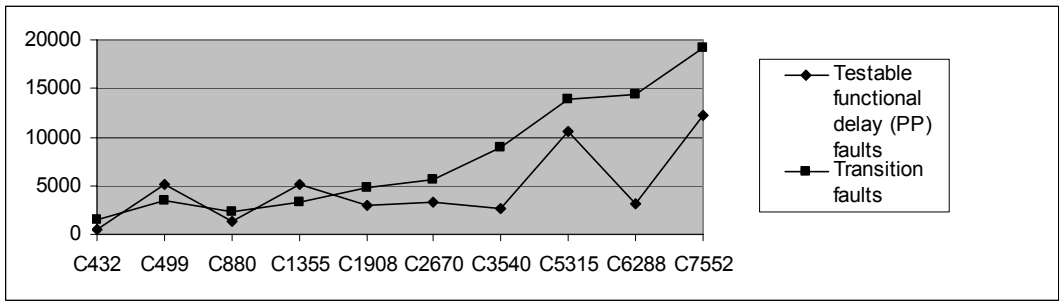

Figure 2. Numbers of testable functional delay (PP) and transition faults

Table 2. Functional delay tests vs. transition tests

Circuit	The application of functional delay fault test for detection of transition faults					The application of transition fault test for detection of functional delay faults				
	Test size 1	Test size 2	Transition faults			Test size 3	Test size 4	Functional delay faults		
			Number	Detected	Coverage 1 (%)			Number	Detected	Coverage 2 (%)
C432	117	610	1412	1288	91.22	290	610	540	510	94.44
C499	1077	10302	3430	3418	99.65	448	7744	5184	3873	74.71
C880	381	1920	2396	2302	96.08	296	1360	1326	1018	76.77
C1355	1011	10292	3350	3317	99.01	574	7670	5184	3845	74.17
C1908	620	4612	4848	4594	94.76	658	2176	3004	1726	57.45
C2670	448	3584	5646	5447	96.48	484	2694	3320	2487	74.90
C3540	515	2954	8960	7533	84.30	830	2506	2588	2348	90.72
C5315	1169	9604	13816	13565	98.18	590	7034	10540	8230	78.08
C6288	268	2064	14422	14386	99.75	236	1858	3068	2872	93.61
C7552	2115	11602	19160	18494	96.52	912	5988	12188	7195	59.03
Average	772	5754			95.59	532	3964			77.38

Test size 1 - size of test for PP faults

Test size 2 - size of test for functional delay faults composed from the test for PP faults according to Rule 2

Test size 3 - size of test for transition faults

Test size 4 - size of test for functional delay faults composed from the test for transition faults according to Rule 2

Coverage 1 - transition fault coverage of test generated for functional delay faults detection

Coverage 2 - functional delay fault coverage of test generated for transition faults detection

The rate of testable functional delay faults ranges from 8% (C2670) to 99% (C1355). There is no obvious correlation between the circuit size and the rate of testable functional delay faults. This rate likely depends only on the function which implements the considered CUT. The numbers of testable functional delay and transition faults are comparable for very small circuits (less than 1000 gates), for bigger circuits the numbers of transition faults are bigger and there is a linear like dependence between the number of transition faults and the circuit size.

The comparison of various test set sizes and detected faults of the non-redundant ISCAS'85 benchmark circuits are given in Table 2 and Figures 3, 4. Note that the functional delay and transition fault test sets detect 100% of targeted faults. Another observation is that the test patterns of transition fault test set were considered as separate patterns, i. e. for each pattern u and v in the pair $<u, v>$, which was generated for detection of particular transition faults, Rule 2 was applied.

If we examine the results of experiments presented in Table 2 and Figure 3, we can see that the test sizes for PP faults (Size 1) and transition faults (Size 3) are of the same range. On the average, the test sizes for PP faults are 1.45 times larger than the test sizes for transition faults, however there are two exceptions, i.e. the circuits c432 and c2670. This trend remains also for functional delay tests obtained from PP tests and transition tests according to Rule 2 (Size 2 and 4, respectively). The functional delay tests obtained from PP tests are on the average 1.45 times larger than the test sizes of functional delay tests obtained from transition tests too. Another interesting sighting is that the

test size enlargement due to transformation of PP and transition tests into functional delay tests is on the average 7.45 in both cases for considered circuits. Thus on the average almost four test pairs are generated for each separate test pattern.

The numbers of detected faults and test coverages are given in Table 2 and Figure 4. The average percent of detected transition faults by the tests generated for the function delay faults exceeds 95.5 %, but the minimum percent of detected transition faults is 84.3 % (circuit c3540). As we see, the test sets, which detect 100% transition faults of the benchmark circuits and are transformed into functional delay tests, detect on the average 77.4 % of the functional delay faults. The rate of functional delay fault coverages of tests generated for transition faults detection ranges from 57.45 % (C1908) to 94.44 % (C432). It is worth to note that these coverages are maximal because there is no tool which could be able to analyse the transition fault test not as separate patterns but as pattern pairs, i. e. there is no tool which can take into account only signal value transitions that take place between test patterns in the pair $<u, v>$. However the transition tests are more suitable for functional delay testing than test sets generated for stuck-at faults. The results presented in [17] show that the test sets, which detect 100% stuck-at faults, detect on the average about 60% of the functional delay faults.

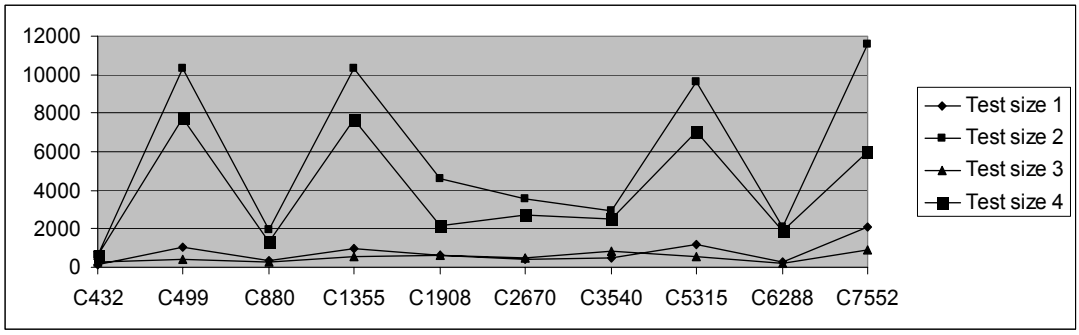

Figure 3. Test sizes

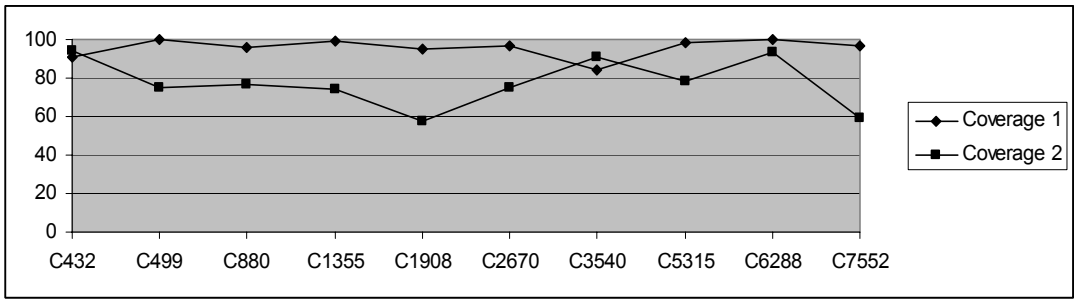

Figure 4. Fault coverages

5. Conclusions

Our experimental results show that the test sets, which are generated according to the functional delay fault model, obtain high fault coverages of transition faults. However, the functional delay fault coverages of the test sets targeted for the transition faults are low. This implies that a test set based on the functional delay fault model covers far much more than the single transition faults. It is very likely that the test vectors based on the functional delay fault model can cover other kinds of the faults. Another advantage of test set generated at the functional level is that it is independent of and effective for any implementation and, therefore, can be generated at early stages of the design process.

References

[1] S.Ohtake, K.Ohtani, H.Fujiwara. A Method of Test Generation for Path Delay Faults Using Stuck-at Fault Test Generation Algorithms. *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition* (DATE'03), 2003, 310-315.

[2] A. Krstic, J.-J. Liou, K.-T. Cheng, L.-C. Wang. On Structural vs. Functional Testing for Delay Faults. *Proceedings of IEEE International Symposium on Quality Electronic Design, March*, 2003, 438-441.

[3] J.A. Waicukauski, E. Lindbloom, B.K. Rosen, V.S. Iyengar. Transition fault simulation. *IEEE Design and Test, April* 1987, 32-38.

[4] V.S. Iyengar, B.K. Rosen, J.A.Waicukauski. On Computing the Sizes of Detected Delay Faults. *IEEE TCAD, March* 1990, 299-312.

[5] C.J. Lin, S.M. Reddy. On Delay Fault Testing in Logic Circuits. *IEEE Transactions on Computer-Aided*

Design of Integrated Circuits and Systems, *Vol*.6, *September* 1987, 694-703.

[6] I. Pomeranz, S.M. Reddy. Test Enrichment for Path Delay Faults Using Multiple Sets of Target Faults. *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition* (DATE'02), 2002, 722-729.

[7] J.-J. Liou, L.-C. Wang, K.-T. Cheng. On theoretical and practical considerations of path selection for delay fault testing. *Proceedings of the* 2002 *IEEE/ACM International Conference on Computer-aided Design*, 2002, *San Jose, California, USA, November* 10-14, 2002, 94-100.

[8] I. Pomeranz, S.M. Reddy. On Generating High Quality Tests for Transition Faults. *Proceedings of the* 11th *Asian Test Symposium* (ATS'02) *November* 18-20, 2002, 1-8.

[9] K. Yang, K.-T. Cheng, L.-C. Wang. TranGen: A SAT-Based ATPG for Path-Oriented Transition Faults. *Proceedings of the 41th Design Automation Conference, DAC* 2004, *San Diego, CA, USA, June* 7-11, 2004, 92-97.

[10] A.K. Majhi, V.D. Agrawal. Tutorial: Delay Fault Models and Coverage. *Proceedings of the Eleventh International Conference on VLSI Design: VLSI for Signal Processing, January* 04-07, 1998, 364.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] E. Bareisa, R. Seinauskas. 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.

[17] E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas. The Realization-Independent Testing Based on the Black Box Fault Models. *Informatica*, 2005, *Vol*.16, *No*.1, 19-36.

[18] F. Brglez, H. Fujiwara. A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran. *IEEE Int'l Symp. on Circuits and Systems*, *Vol*.3, *June* 1985, 663-698.