ISSN 1392 - 1215 ELEKTRONIKA IR ELEKTROTECHNIKA. 2005. Nr.3(59)

T 170 ELEKTRONIKA

Transition Fault Coverage For Different Implementations Of The Circuit

E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas

Software Engineering Department, Kaunas University of Technology
Studentų St. 50-406, Kaunas, Lithuania
e-mail: eduardas.bareisa@ktu.lt, vacius.jusas@ktu.lt, kestutis.motiejunas@ktu.lt, rimantas.seinauskas@ktu.lt

1. Introduction

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. Tests can be generated for a high level description in order to reuse them for all possible implementations [1]. However, such tests usually cannot guarantee the detection of all specified faults in all possible implementations. Consequently, if we consider realization-independent testing, we can only speak about such realizations that fulfil specific requirements or have a particular structure [2, 3].

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 [4]. Increasing performance requirements of circuits makes 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.

In this paper we will analyse the situation when tests are generated for a particular implementation. In this case there naturally rises a question – can a test generated for one implementation be used for another implementation? The same core can have distinct descriptions; e. g. a parallel or sequential carry can be realized in an adder. Naturally, that a test generated according to one structure may not detect all specified faults of another structure. The employment of different synthesis tools can have an influence on the test quality as well. The problems of generation of realization-independent tests for stuck-at faults were addressed in [5, 6, 7]. We will investigate the delay faults coverage in various implementations of the same circuit.

In this work we will analyse such implementations that are generated by the synthesis tool according to the same description, changing the synthesis tool and the target library used during the synthesis. We will explore the test quality of one realization for detecting faults of other realizations. The ISCAS'85 benchmark circuits will be used for experiments. As well we will analyse how the tests for delay faults can be modified or expanded in order to enhance the fault coverage of other realizations and we will evaluate such possibilities by experiment.

The conventional synthesis goal is to find a trade-off between the minimal area and the maximal performance. The different implementations could be based on these extremities: low area and high speed [5]. We have tried to synthesize the circuits targeted on the low area and the high speed. But the obtained results were very similar. Then we changed a target library. The obtained results of different target libraries were quite different. Therefore, the choice was made for the implementations based on the different target libraries.

The structure of the paper is as follows. We review the related work in Section 2. We analyse the influence of circuit re-synthesizing on the transition fault coverage in Section 3. We explore the application of transition faults tests to detect stuck-at faults in Section 4. We present the enhancement of the independency of the test from realizations in Section 5. We finish with conclusions in Section 6.

2. Related work

Two general types of delay fault models: the gate delay fault model [8, 9], and the path delay fault model [10] 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 [8]. 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.

The possibilities of using a test obtained for one realization for testing delay faults of another realization are studied in [5]. The suggested fault model is called a coupling fault, which is devoted to testing stuck-at faults and is applicable to test path delay faults. The corresponding coupling delay tests detect all robust path delay faults in any realization of the function. The size of a coupling delay test set is very large compared to that of a typical path delay test set, however [5].

The possibilities of supplementing or expanding a particular realization test having a purpose to enhance test quality for detecting of delay faults are analysed in [11-14]. 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 path. This may lead to undetected failures since a shorter path may fail without any of the longest paths failing. The paper [11] proposes 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. The alternative approach to this problem is an optimisation of the critical path selection [15] or a selection of the longest testable path [12, 16]. The papers [12, 16] combine the merits of both the transition fault model and the critical path delay model. 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 the paper [12] suggests that 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.

The other possibility to enhance test quality is the n-detection test set [13, 14]. The *n*-detection test set is one where each fault *f* is detected by *n* different input patterns, or by the maximum number of input patterns if *f* has fewer than *n* different input patterns that detect it. The paper [13] has proposed a reordering procedure to obtain n-detection test sets and variable n-detection test sets for transition faults. Though t_i and t_{i+1} are selected from the given test set as a test-pair for transition faults, authors do not consider the number of input changes between t_i and t_{i+1}. However, the multiple input change test-pairs have the following disadvantages: 1) hazards may occur by multiple input change test-pairs, and 2) multiple input change test-pairs have a high power consumption. Further, the authors in [17] proved that single input change test sequences are more effective than multiple input change sequences to obtain high robust delay fault coverage. The paper [14] applies n-detection test sets to check path delay faults where n is a function of the number of paths through the check points.

3. Application of transition fault tests to re-synthesized circuits

The core can be synthesized by different electronic design automation systems and mapped into different cell libraries and manufacturing technologies. An important issue is how the test set of the core covers the transition faults of new implementations, which are done by the same synthesizer or by a different one. The ISCAS'85 benchmark circuits have been selected for experiments. Two versions of the original circuits have been taken: the first original version and the second improved non-redundant version. The first original version of the circuits was considered as the basic version. Unfortunately, to use this version as the basic was possible only for five circuits: C432, C499, C880, C1355, C6288. The first version of the other circuits couldn't be used for two reasons: 1) the versions of the circuits C1908, C3540 and C5315 had different interface; 2) the first versions of the circuits C2670 and C7552 couldn't be compiled because of the applied restrictions by the Synopsys test pattern generator TetraMAX used for our experiments. The original non-redundant ISCAS'85 circuits have been re-synthesized by the Synopsys Design Compiler program in three modes and by the Cadence BuildGates synthesis program. The Synopsys Design Compiler program used three different target libraries: 1) class.db – default mode; 2) and_or.db – AND-OR-NOT cell library of two inputs gates; 3) virtex.db – FPGA cell library. The following six realizations have been analyzed:

V0 – the original ISCAS'85 benchmark circuit,
V1 – the non-redundant ISCAS'85 benchmark circuit,
V2 – Synopsys Design Optimization,
target library – class.db,
V3 – Synopsys Design Optimization,
target library – and_or.db,
V4 – Synopsys Design Optimization,
target library – virtex.db,
V5 – Cadence BuildGates synthesis,
target library – lca300k.alf.

Table 1. The number of transition faults

Circuit	V0	V1	V2	V3	V4	V5
C432	1438	1412	1002	1172	1228	1050
C499	3436	3430	2646	2982	3138	2646
C880	2396	2396	2146	2280	3040	2170
C1355	3366	3350	3274	3618	3306	3274
C1908	4872	4848	2176	2796	2996	2440
C2670	-	5646	4134	4486	4922	4162
C3540	9360	8960	6154	6448	6942	8024
C5315	13988	13816	10312	10364	13382	10652
C6288	14560	14422	13528	14790	18180	25678
C7552	-	19160	11962	12048	14136	12898
Total	53416	77440	57334	60984	71270	72994

We can see the number of transition faults for each realization in Table 1 and 0. It needs to draw attention to the fact that originally the circuits C432 and C499 have XOR gates. When a test is generated for XOR gate, it does not detect 2 faults of the equivalent circuit constructed of NOT, AND and OR gates. All the other versions of the circuits are constructed from primitive gates: AND, OR,

NAND, NOR and NOT. In order to have equal conditions for all versions of circuits, the original circuits C432 and C499 were expanded to the NOT, AND and OR gates. The version V5 of the circuits is constructed from FPGA cells. The transition faults of these circuits were simulated at the gate level, too.

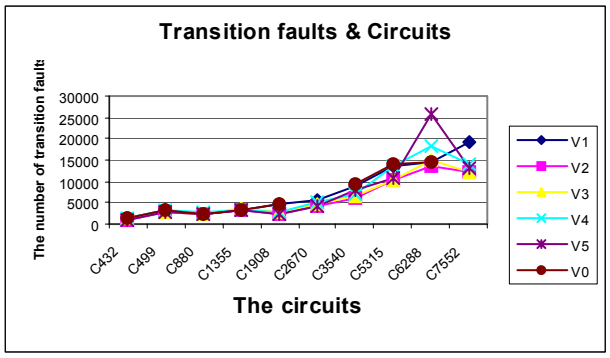

Fig. 1. The number of transition faults for each realization

The original non-redundant benchmark realizations have more transition faults in total. This means that the re-synthesized circuits were more optimised. The percent of the difference between maximum and minimum numbers of transitions faults to the maximum number of transition faults varies from 10 to 55. It demonstrates the diversity of realizations and the impact of the target library on the design synthesis.

The number of transition faults has to be equal to the number of stuck-at faults because every pin of the gate has two transition faults (a slow-to-rise fault and a slow-to-fall fault) and two stuck-at faults (stuck-at 0 and stuck-at 1). If we look to the paper [7] which presented the numbers of stuck-at faults for benchmarks circuits we would see two or three times smaller numbers. The reasons are the following: 1) a simulation program of the stuck-at faults for two inputs AND gate includes 4 stuck-at faults into a fault list meanwhile a simulation program of the transition faults for the same gate includes all 6 transitions faults; 2) the equivalence and dominance relations are applied for stuck-at faults. Therefore, for example, the circuit C880 has 942 stuck-at faults [7] and 2396 transition faults.

Synopsys test pattern generator TetraMAX was used to generate test sets for transition faults. The test sets have been generated for each basic version of the original ISCAS'85 circuit and then the same test set was applied to all the other implementations of the same circuit. The choice of the basic version for each circuit was discussed in the beginning of this section. As we remember two original versions of the circuits C1908, C3540 and C5315 had different interface. Therefore, two test sets were generated for these circuits. The test sets of the version V0 were used only for this particular implementation. The test size of test sets with a 100% transition faults coverage is displayed in Table 2. In each test generation case, we see the test size dispersal (0) in the number of circuits. The test sizes for the realisation of the circuits c432, c880, c6288 are very similar. Nevertheless, these circuits have quite a different dispersal of transition faults after re-synthesizing (Table 1).

Table 2. The size of test sets

	C432	C499	C880	C1355	C1908
V0	268	434	282	620	630
V1					628
	C2670	C3540	C5315	C6288	C7552
V0		838	580	236	
V1	510	836	598		912

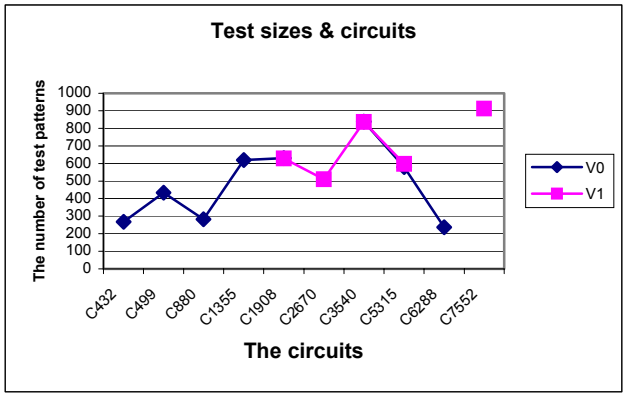

Fig. 2. The number of test patterns for each circuit

The fault coverage and the number of undetected faults for each realization of the circuit were computed. Table 3 presents the results of the experiments. Two lines are reserved for each circuit. The first line of these two lines holds the number of undetected transition faults, the second line holds the transition fault coverage. Of course, the number of undetected faults is zero for the version V0 of all circuits. The version V1 differs very slightly from the version V0 of all circuits and therefore the faults of this version were detected very well, but we don't have to forget that this version was the basic one for 5 circuits, too. The faults of all the other versions of the circuits were not detected completely, except the version V3 of the circuit C432. In general, the version V3 was checked the best for all the circuits. Recall, that this version is implemented on the base of two inputs gates. The worst result has the version V5, but the only circuit C6288 had a big contribution to this result (see 0). From the engineering point of view, this result could be excluded. Then the least fault coverage is 93.06% of the version V4 of the circuit C880. This result leads to the conclusion that different synthesis tools have no real influence to the detection of the transition faults. If we compare the results of the stuck-at faults for the other realizations presented in the paper [7], we could find that the biggest average percent of undetected faults is 1.35. The biggest average percent of undetected transition faults from Table 3 is 2.53. But this result includes the worst case of the circuit C6288 which could be excluded. If the result of the circuit C6288 is excluded, then the biggest average percent of undetected transition faults is 1.81. But when the comparison is accomplished an attention has to be paid to the fact that stuck-at faults were exercised only for two implementations: V2 and V3. The biggest average percent of undetected transition faults of these implementations is 1.36 which means the same as for stuck-at faults.

Table 3. The undetected transition faults and a fault coverage

Circuit	V0	V1	V2	V3	V4	V5
C432	0 100	36 97.45	26 97.41	0 100	46 96.25	26 97.52
C499	0 100	0 100	13 99.51	1 99.97	8 99.75	13 99.51
C880	0 100	0 100	1 99.95	1 99.96	211 93.06	1 99.95
C1355	0 100	0 100	128 96.09	111 96.93	36 98.91	128 96.09
C1908	0 100	0 100	24 98.90	17 99.39	19 99.37	24 99.02
C2670		0 100	114 97.24	130 97.10	28 99.43	114 97.26
C3540	0 100	0 100	52 99.16	63 99.02	50 99.28	315 96.07
C5315	0 100	0 100	40 99.61	64 99.38	429 96.79	52 99.51
C6288	0 100	10 99.93	127 99.06	77 99.48	145 99.20	2430 90.54
C7552		0 100	64 99.46	62 99.49	26 99.82	77 99.40
Total	0	46	589	526	998	3180
%	100	99.74	98.64	99.07	98.19	97.49

Table 4. The undetected stuck-at faults and a fault coverage

Circuit	V0	V1	V2	V3	V4	V5
C432	0 100	36 97.45	5 99.50	0 100	30 97.56	5 99.52
C499	0 100	0 100	0 100	0 100	0 100	0 100
C880	0 100	0 100	0 100	0 100	149 95.10	0 100
C1355	0 100	0 100	58 98.23	46 98.73	8 99.76	58 98.23
C1908	0 100	0 100	3 99.86	1 99.96	1 99.97	3 99.88
C2670		0 100	73 98.23	81 98.19	6 99.88	73 98.25
C3540	0 100	0 100	1 99.98	5 99.92	2 99.97	139 98.27
C5315	0 100	0 100	15 99.85	14 99.86	336 97.49	19 99.82
C6288	0 100	0 100	37 99.73	25 99.83	39 99.79	1453 94.34
C7552		0 100	6 99.95	2 99.98	3 99.98	18 99.86
Total	0	36	198	174	574	1768
%	100	99.75	99.53	99.65	98.95	98.82

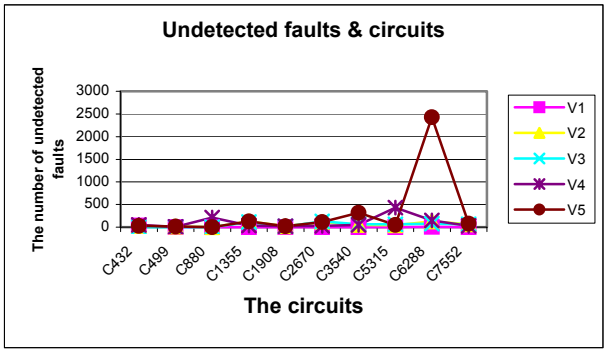

Fig. 3. The total number of undetected transition faults for each realization of the circuit

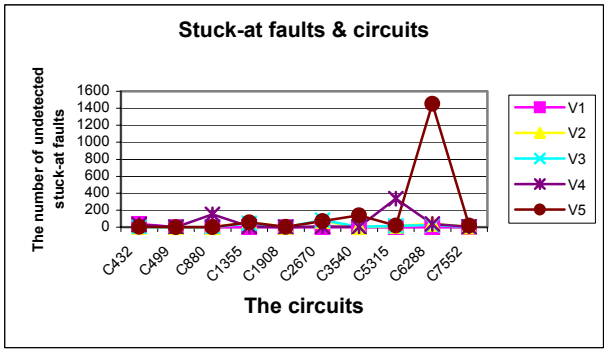

Fig. 4. The total number of undetected stuck-at faults for each realization of the circuit

4. Application of transition fault tests to detect stuck-at faults

The generated test sets for transition faults were applied to all implementations of the circuits in order to detect stuck-at faults. The results of the fault coverage and undetected faults are presented in Table 4. Two lines are reserved for each circuit. The first line of these two lines holds the number of undetected transition faults, the second line holds the transition fault coverage. The total number of stuck-at faults was the same as reported in Table 1. As we could expect the detection of stuck-at faults is better than their counterparts transition faults. To check the trends of the detection of stuck-at faults for different circuits and for different implementations and to compare them with trends of the detection of the transition faults one needs to look at the 0 and 0. As we can see the distribution law is absolutely the same only the numbers of undetected stuck-at faults are smaller than their counterparts transition faults. It is possible to conclude that the trends of the detection of stuck-at faults and transition faults for different implementations are absolutely the same.

5. Enhancement of the independency of the test from realizations

As it is reported in [13, 14], *n*-detection test sets are useful in achieving a higher defect coverage for all types of circuits and for different fault models. We applied merged test sets for testing as a double-detection approach. The single test set consists of a random generated test set and a deterministic generated test set. The deterministic test generation was used only for hard-to-detect faults. It contributed very few test patterns to the final test set. Therefore both test sets have different test patterns and each of them detects all faults of the basic realization (V0 or V1). The numbers of test patterns of both test sets are presented in the second column of Table 5. The first number represents the number of test in the first set, the second number - the number of test in the second set. The numbers of undetected faults of double-detection test sets and a fault coverage are given in the other columns of Table 5. The column under name V0 has empty cells for the circuits C1908, C2670, C3540, C5315, C7552, because the basic version of these circuit was V1.

Table 5. The fault coverage

Circuit	V0/ V1	V0	V1	V2	V3	V4	V5
C432	268 264	0 100	36 97.5	5 99.5	0 100	35 97.2	5 99.5
C499	434 420	0 100	0 100	10 99.6	0 100	4 99.9	10 99.6
C880	282 302	0 100	0 100	0 100	0 100	171 94.4	0 100
C1355	620 646	0 100	0 100	2 99.9	2 99.9	4 99.9	2 99.9
C1908	628 612		0 100	5 99.8	16 99.4	5 99.4	5 99.8
C2670	510 532		0 100	85 97.9	80 98.2	6 99.9	85 98.0
C3540	836 832		0 100	11 99.8	14 99.8	13 99.8	158 98.0
C5315	598 584		0 100	16 99.8	17 99.8	367 97.3	52 99.5
C6288	236 236	0 100	2 99.9	18 99.9	7 99.9	20 99.9	1286 95.0
C7552	912 894		0 100	13 99.9	17 99.9	12 99.9	21 99.8
Total		0	38	165	153	637	1624
%		100	99.7	99.6	99.7	98.8	98.9

The average percent of undetected faults in case of double-detection test sets declined more than twice (0), except the version V4 of circuits. Also the maximum percent of undetected faults has the same tendency as the average percent of undetected faults, except the version V2 of circuits (0).

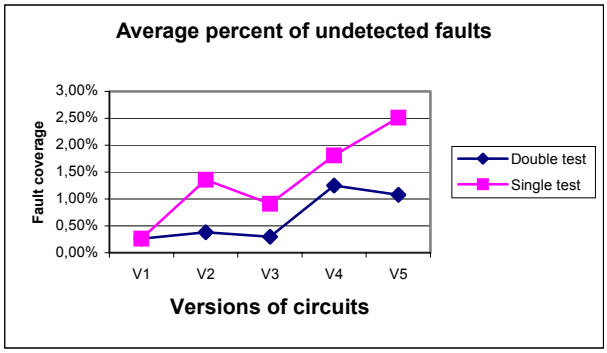

Fig 5. The average percent of undetected transition faults for each realization of the circuit

The version V1 of circuits is a special case. For some circuits (C1908, C2670, C3540, C5315, C7552), it was the basic one. Therefore the single test set has no undetected transition faults for the version V1 of these circuits. For other circuits (C499, C880, C1355), the single test set already detects all transition faults for the version V1. And it is very interesting to notice, that the double-detection test sets did not improve the test quality for the version V1 of the circuit C432. That was the only case where double-detection test sets did not increase the coverage of undetected faults of a single test set. The latter fact only reminds to us that double-detection test sets are not enough in order to improve a test quality. In general, the maximum percent of undetected faults 9.46% (0) is significantly higher than the average percent of undetected faults 2.51% (0). The double-detection test sets decreased almost twice both the maximum and the average percent of undetected faults.

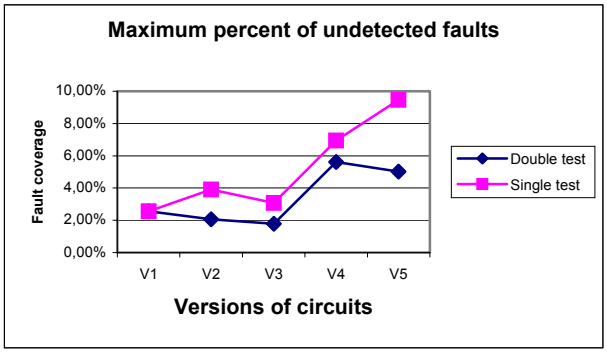

Fig. 6. The maximum percent of undetected transition faults for each realization of the circuit

6. Conclusions

The comparison of the detection of the transition faults for different implementations of the circuit was carried out the first time. The results show that the tests reused for re-synthesized circuits detect on average more than 98% of all transition faults. The maximum percent of undetected faults 9.46% is significantly higher than the average percent of undetected faults 2.51%. The double-detection test sets declined the maximum and the average percent of undetected faults almost twice.

The comparison of the trends of the detection of stuck-at faults for different implementations [7] and the detection of transition faults for different implementations was carried out, too. It is possible to conclude that the trends of the detection of stuck-at faults and transition faults for different implementations are the same. Finally, the presented results lead to the conclusion that re-synthesizing of a circuit doesn't create any problems to the testing of re-synthesized circuit – it is possible to apply the test patterns of the original circuit for the re-synthesized one with a little loss in a fault coverage for the re-synthesized circuit.

References

1. **Y. Zorian, S. Dey, and M. Rodgers**. Test of Future System-on-Chips. Proceedings of the 2000 International Conference on Computer-Aided Design, November, 2000. – P. 392-398.
2. **S. B. Akers**, Universal Test Sets for Logic Networks, IEEE trans. Computers, vol. C-22, 1973. – P. 835-839.
3. **R. Betancourt**, Derivation of Minimum test sets for Unate Logic Circuits, IEEE Trans. Computers, Vol. C-20. – Nov. 1971. – P.1264-1269.
4. **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. – P.310-315.
5. **J. Yi and J.P. Hayes**, A Fault Model for Function and Delay Testing, Proc. of the IEEE European Test Workshop, ETW'01, 2001. – P. 27-34.
6. **H. Kim and J. P. Hayes**, Realization independent ATPG for designs with unimplemented blocks, IEEE Transactions on CAD, Vol. 20, No. 2, February 2001. – P. 290-306.

7. **E. Bareiša, V.Jusas, K.Motiejūnas, R.Šeinauskas**, The Influence of Circuit Re-Synthesizing on The Fault Coverage, Information Technology and Control, ISSN 1392-124X. Kaunas, Technologija, 2004. – Nr. 2 (31). – P.7-15.
8. **J.A. Waicukauski, E. Lindbloom, B.K. Rosen and V.S. Iyengar**, Transition fault simulation, IEEE Design and Test, April 1987. – P.32-38.
9. **V.S. Iyengar, B.K. Rosen and J.A.Waicukauski**, On Computing the Sizes of Detected Delay Faults, IEEE TCAD, March 1990. – P.299-312.
10. **C.J.Lin and 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. – P.694-703.
11. **I. Pomeranz and 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. – P.722-729.
12. **E. Bareiša, V.Jusas, K.Motiejūnas, Š.Packevičius, R.Šeinauskas**, The Improvement of Test Independence from Circuit Realization, Information Technology and Control, ISSN 1392-124X. Kaunas, Technologija, 2004, Nr. 4 (33). – P.45-52.
13. **I. Pomeranz and S. M. Reddy**, On n-Detection Test Sets and Variable n-Detection Test Sets for Transition Faults, in Proc. 17 th VLSI test Symp. , April 1999. – P. 173-179.
14. **H. Takahashi, K.K. Saluja, Y. Takamatsu**, An Alternative Method of Generating Tests for Path Delay Faults Using N_i - Detection Test Sets, In Proc. of the 2002 Pacific Rim International Symposium on Dependable Computing (PRDC'02), 2002. – P. 275-282.
15. **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. – P.94-100.
16. **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. – P.92-97.
17. **E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas**. Testing of FPGA Logic Cells. Electronics and electrical engineering, ISSN 1392-1215, Kaunas, Technologija, 2004, No. 7(56). – P. 37 – 42.

Pateikta spaudai 2005 01 14

E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas. Vėlinimo gedimų apimtis esant įvairioms schemos realizacijoms // Elektronika ir elektrotechnika. - Kaunas: Technologija, 2005. - Nr.3(59). – P.78–83.

Projektuojant lustų sistemas, naudojami iš anksto paruošti blokai, kurių ventilinio lygmens sudarymo detalės yra nežinomos. Šių blokų testai priklauso nuo gamybos technologijos ir kiekvieną kartą gali keistis. Straipsnio tikslas – padėti projektuotojui priimti sprendimą dėl tokių blokų vėlinimo gedimų testavimo. Atlikus daug eksperimentų su standartinėmis kombinacinėmis schemomis, nustatyta, kad vienos realizacijos testai neaptinka vidutiniškai tik 1.5% vėlinimo gedimų persintezuotose kitose tos pačios schemos realizacijose, nors kai kuriais atvejais neaptinkama per 9% gedimų. Tos pačios tendencijos būdingos ir konstantiniams gedimams. Testai, du kartus aptinkantys gedimus, beveik du kartus sumažina neaptinkamų gedimų skaičių visoms schemoms, išskyrus vieną vienos schemos realizaciją. Il. 6, bibl. 17 (anglų kalba; santraukos lietuvių, anglų ir rusų k.).

E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas. Transition Fault Coverage for Different Implementations of the Circuit // Electronics and Electrical Engineering. - Kaunas: Technologija, 2005. – No 3(59). – P.78–83.

The design complexity of systems on a chip drives the need to reuse legacy or intellectual property cores, whose gate-level implementation details are unavailable. The core test depends on manufacturing technologies and changes permanently during a design lifecycle. The purpose of this paper is to assist to designer in the decision making how to test transition faults of re-synthesized cores. We have performed various comprehensive experiments with combinational benchmark circuits. Our experiments show that the test sets generated for a particular circuit realization fail to detect in average only less than 1.5% of the transition faults of the re-synthesized circuit but in some cases this figure is more than 9%. The same trends are valid for stuck-at faults of different implementations, too. The double-detection test sets declined almost twice both the maximum and the average percent of undetected transition faults for all implementations of the circuits, except one singular implementation of one circuit. Ill.6, bibl. 17 (English, summaries in Lithuanian, English and Russian).

Э. Барейша, В. Юсас, К. Мотеюнас, Р. Шейнаускас. Анализ полноты дефектов задержки для разных реализаций схемы // Электроника и электротехника. - Каунас: Технология, 2005. – № 3(59). – С.78–83.

При проектировании сложных современных систем используются уже готовые блоки, детали которых на вентильном уровне являются неизвестными. Тест для такого блока зависит от технологии изготовления и может каждый раз меняться. Цель этой работы помочь проектировщику принять решение по поводу тестирования дефектов задержки таких блоков. Мы провели много экспериментов, используя комбинационные схемы. Наши эксперименты показывают, что тесты одной реализации в среднем не обнаруживают только 1,5% дефектов задержки других реализаций, однако в некоторых случаях эта цифра равна 9%. Эта же самая тенденция соблюдается и для константных одиночных неисправностей. Тесты, обнаруживающие неисправности по два раза, уменьшили средний и максимальный процент необнаруживаемых неисправностей в два раза, за исключением одной схемы. Ил. 6, библ. 17 (на английском языке; рефераты на литовском, английском и русском яз.)

DOI: 10.5755/j02.eie.10377