ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 *2010. No. 9(105)*
ELEKTRONIKA IR ELEKTROTECHNIKA

ELECTRONICS
T 170
ELEKTRONIKA

Factor of Randomness in Functional Delay Fault Test Generation for Full Scan Circuits

E. Bareiša, P. Bieliauskas, V. Jusas, A. Targamadzė
Software Engineering Department, Kaunas University of Technology, Studentų str. 50-406, Kaunas, Lithuania
L. Motiejūnas
Multimedia Engineering Department, Kaunas University of Technology, Studentų str. 50-401A, Kaunas, Lithuania
R. Šeinauskas
Information Technology Development Institute, Kaunas University of Technology, Studentų str. 48A, Kaunas, Lithuania

Introduction

The functional test development at a high level of abstraction is an important direction of the test generation. In the initial stages of the design, the structural implementation of the design is unknown. But the functional test development can be accomplished in parallel with other design stages. In this case, the time of test generation is not a critical issue.

The generation of the functional test to detect delay faults has a long history [1, 2]. According to the terminology introduced in [1], a functional delay fault is a tuple (*I*, *O*, *trI*, *trO*), where *I* is an input of the circuit under test (CUT), *O* is an output of the CUT, *trI* is a rising or falling transition at *I*, and *trO* is a rising or falling transition at *O*. 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 the circuit. Underwood et al. [2] proposed the method that was devoted to generate functional tests for standard scan designs using functional justification approach. Now, the more accepted term for this type of test is a broadside test [3]. Pomeranz and Reddy [1] proposed the methods that were devoted to the combinational circuits only.

The presented methods [1, 2] do not consider the whole circuit as a single functional block, and they have the problems in handling large functional blocks. Several black-box fault models suggested later in [4–6] have no such problems. The single coupling fault model [6] is defined in terms of a single input/output pair. The average size of the test set is 2^n - 1, where *n* denotes the number of inputs of the module [6]. Therefore, the test sets are very large even for small modules and not practical in many cases.

The coupling delay tests [6] are the single input transition (SIT) tests. Jusas and Motiejūnas [5] introduced the functional delay multiple input transition (MIT) tests. The approach was extended in [4]. In this approach, the delay faults are detected in the function-robust and function-non-robust manner. Such a versatility of the approach allows obtaining high transition fault coverage and quite a small number of delay test patterns. The proposed methods [4, 5] were applied for the combinational circuits only.

The purpose of this paper is to explore the influence of random generation methods to the results of functional delay test generation for full scan circuits. We cover the necessary background on random generation in the next section.

Background

Functional testing is a form of black box testing, which does not require the structural description of the circuit under test. It avoids the problem of deterministic test generation using structural information about the circuit under test. Available evidence [1, 4, 7, 8] suggests that random or pseudorandom testing may be reasonable choice for functional test pattern generation. The test patterns are generated randomly, and then some patterns, which satisfy appropriate selection criteria based on the functional model of the circuit, are selected and included into the final test set. Such the test patterns are already called the functional test patterns.

Random test pattern generation does not exploit some information that is available in black box testing environment. This information consists of the previous tests applied. If an experienced engineer is writing tests manually, he would select each new test such that it covers some part of the functionality not yet covered by the tests already written. Random test pattern generation, which explicitly uses the information about the past patterns applied for generating a new pattern, is called anti-random test pattern generation [9–11]. Malaiya [9] introduced the

concept of anti-random testing. The approach is based on the hypothesis that if two input patterns have only small distance between them, then the sets of faults covered by the two are likely to have a number of faults in common. Therefore, the basic premise of anti-random testing is to choose new test patterns that are as far away from existing test inputs as possible. The idea of anti-random testing was differently implemented in [10] and [11], which is the extension of [9]. The main deficiency of [10] was that anti-random test generation saturates too soon and it can not provide the highest coverage in every case. The concept of balanced space was used as a natural stopping criterion.

The anti-random test generation presented in [11] has the following deficiencies: 1) it uses exhaustive search; 2) it is not adjusted to stop generation according to some criterion. The one characteristic is common for both papers [10] and [11] – they are not targeting the delay faults.

We will adapt the anti-random test generation for functional delay faults and we will look at the possibilities of random test generation in the next section.

Random generation

A random generation of values 0 and 1 are used for the formation of the pairs of the functional delay test patterns. The random generation can be accomplished in three different ways:

- random generation of values 0 and 1 when the probability for the appearance of each value is 50% in every bit of test pattern. We denote such a mode of random generation by “Rand1”;
- random generation of large integer values, which then are unfolded into the sequences of 0’s and 1‘s according to the rules of the conversion of the decimal number to the binary number. In this case, the probability of the appearance of the values 0 and 1 is not known in every bit of the test pattern. The precaution has to be taken. The highest bits of the converted binary number should not be used, because they as usually hold values 0. Such a result is related to the generation algorithm of random integer values. They are generated evenly distributed through the whole period between value 0 and the maximal value. The values close to the maximal value appear quite rarely. We denote such a mode of random generation by “Rand2”;
- anti-random generation using the ideas presented in [9–11]. We will present the approach of the anti-random generation for the functional delay test patterns. We denote such a mode of random generation by “AntiR”.

The anti-random generation is applied for the detection of stuck-at faults when every separately considered test pattern detects stuck-at faults. The basic premise of anti-random generation is to choose a new test pattern that is far away from existing test inputs as possible. The detection of delay faults requires two test patterns. The first pattern sets the values; the second pattern launches the transitions. So, the transitions, which depend on both patterns, enable the detection of delay faults. To generate both test patterns separately according to the ideas of anti-random generation there is no meaning, because the detection of the delay faults depends on the transitions between both test patterns. In order to navigate the random search space as largely as possible, the first pattern of the pair is more important, because it sets the state, from which the transitions are launched. Therefore, the first pattern of the pair will be generated using the ideas of the anti-random generation. The second pattern will be generated using the random generation named “Rand2”. In every case, when the random generation has to be used, we use random generation named “Rand2”, because to our belief it is better than random generation named “Rand1”. This belief will be confirmed by the experiments provided in the next section.

Now, we can summarize the ideas that we are going to implement in the anti-random generation for the functional delay fault detection.

The first pair of test patterns is generated randomly. They both are included into the set that will accumulate the patterns for anti-random pattern calculation. The centroid pattern is calculated according to the methodology presented in [10]. Then it is complemented and becomes the first pattern of the pair. The second pattern of the pair is generated randomly. Unlike in [10], both patterns are included into the set that accumulates the patterns for anti-random pattern calculation. The randomly generated pattern is included for the anti-random pattern calculation, because the space without such a pattern balances quite early and it does not allow the further generation. The randomly generated pattern serves as the mutant for the space and it never allows for the space to be balanced.

Experimental results

The experiments were carried out on the circuits of the benchmark suite ITC’99. The results of the experiments are provided in Table 1, Table 2, and Table 3. The first column of the tables holds the circuit name. We provide also the number of inputs and outputs in the first column of Table 1. The number of inputs and the number of outputs includes the number of state bits as well. These numbers show the size, the complexity of the circuit. The second column shows the mode of the random generation. The column under name “Selected” presents the number of selected test inputs. The time of test generation is provided in the column under name “Time”. The time is expressed in minutes. We have used the computer Intel Pentium 4 CPU 3.20GHz. The penultimate column shows the size of the last set of generated test stimuli. The last column under name “TFC” presents the transition fault coverage of selected test stimuli. We have used TetraMAX program to obtain the transition fault coverage.

Table 1 reports the results when the size of the set of generated test stimuli was the same for the circuit in all three modes of generation. The generation was accomplished in the following way. The initial size of the set of generated test stimuli was always 100. If some test patterns were selected from the generated set, the size of the set was not increased for the new iteration. If no test patterns were selected from the generated set, the size of the set was doubled for the new iteration. In such a way, the generation was carried out several times for the same size of the generated stimuli. We stopped the test

generation according to the functional test termination condition when the last iteration selected no new test patterns. This number was not the same for different modes of generation. We stopped the generation according to the mode that reached firstly the termination condition. These modes are shown in bold in the second column, expect the circuit S38417. We stopped the generation for this circuit due to the long generation time.

The randomly generated test patterns "AntiR" obtained the highest test coverage for all the circuits, except circuit S38417, where it differs very slightly. But the time of this generation was quite longer in many cases. Therefore, we provide Table 2, where the time was allotted longer for the random generations that obtained the worse results of transition fault coverage.

Table 1. Comparison of the random generation, when the size of the initial set is the same

Circuit	Mode	Selected	Time, min	Size of last set	TFC (%)
S1196 I - 32 O – 32	Rand1	515	344	10240000	88.84
	Rand2	314	209	10240000	90.44
	AntiR	317	233	10240000	91.55
S1238 I - 32 O - 32	Rand1	490	134	2560000	90.41
	Rand2	310	87	2560000	90.41
	AntiR	320	82	2560000	91.30
S13207 I - 700 O - 790	**Rand1**	2271	610	6400	94.70
	Rand2	3401	1036	6400	97.01
	AntiR	3524	1503	6400	97.41
S15850 I - 611 O - 684	**Rand1**	2779	704	3200	95.67
	Rand2	4065	1540	3200	97.72
	AntiR	3857	1176	3200	97.79
S35932 I - 1763 O - 2048	Rand1	227	56	400	100
	Rand2	226	53	400	100
	AntiR	224	59	400	100
S38417 I - 1664 O - 1742	Rand1	1015	396	200	84.35
	Rand2	9362	2644	200	98.53
	AntiR	9062	2590	200	98.37

Table 2. Comparison of the random generation, when the long generation time is allotted

Circuit	Mode	Selected	Time, min	Number of generated	TFC (%)
S1196	Rand2	314	253	20480000	90.44
	AntiR	317	233	10240000	91.55
S13207	Rand1	2326	1769	25600	94.78
	Rand2	3643	2720	12800	97.30
	AntiR	3524	1503	6400	97.41
S15850	Rand1	2938	2087	25600	95.77
	AntiR	3857	1176	3200	97.79
S38417	Rand1	2458	3791	1600	86.96
	AntiR	9062	2590	200	98.37

We see from Table. 2 that the results were improved little bit but the allotted longer generation time did not allow out running earlier obtained the best result of "AntiR". The obtained outcome underlines the value of the chosen algorithm for the random generation.

The number of selected test patterns is important in many cases as well. Therefore, we provide Table 3, which shows the transition fault coverage when the number of selected test patterns was equalled to the least selected amount of test patterns (see Table 1). Again, as we see, the anti-random generation obtained the highest transition fault coverage for all the circuits, except circuit S13207.

Table 3. Comparison of the random generation, when the size of the final set is the same

Circuit	Mode	Selected	Time, min	Number of generated	TFC, %
S1196	Rand1	314	10	20000	84.34
	Rand2	314	209	10240000	90.44
	AntiR	314	233	10240000	91.55
S1238	Rand1	310	10	20000	84.63
	Rand2	310	87	2560000	90.41
	AntiR	310	57	1280000	91.30
S13207	Rand1	2271	610	6400	94.70
	Rand2	2271	259	400	95.63
	AntiR	2271	205	400	95.37
S15850	Rand1	2779	704	3200	95.67
	Rand2	2779	504	800	96.21
	AntiR	2779	280	800	96.42
S38417	Rand1	1015	396	200	84.35
	Rand2	1015	271	100	92.23
	AntiR	1015	262	100	92.37

Despite the view of consideration of the obtained results, the random generation named "AntiR" is the best choice in every case. It also could be noticed that the random generation named "Rand2" is almost as good as the random generation "AntiR".

Conclusions

We investigated the influence of the random generation methods to the results of the functional delay test generation. There are the three possibilities: random generation of values 0 and 1 when the probability for the appearance of each value is 50% in every bit of test pattern; random generation of large integer values, which then are split into the sequences of 0's and 1's according to the rules of the conversion of the decimal number to the binary number; anti-random generation when the presence of the earlier generated patterns is taken into account. We adopted the anti-random generation for the functional delay faults.

The obtained results of the investigation indicate that the random generation has the direct influence to the results of the functional delay test generation. The highest coverage of transition fault is obtained when the anti-random generation is employed.

References

1. **Pomeranz I., Reddy S. M.** Functional test generation for delay faults in combinational circuits // Proceedings of International Conference on Computer-Aided-Design, 1995. – P. 687–694.
2. **Underwood B., Law W. O., Kang S., Konuk H.** Fastpath: A Path-Delay Test Generator for Standard Scan Designs // Proceedings of International Test Conference, 1994. – P. 154–163.
3. **Bareiša E., Jusas V., Motiejūnas K., Šeinauskas R.** Functional Test Transformation to Improve Compaction // Electronics and Electrical Engineering. – Kaunas: Technologija, 2008. – No. 3(83). – P. 53–58.
4. **Bareisa E., Jusas V., Motiejunas K., Seinauskas R.** Functional Delay Test Generation Based on Software Prototype // Microelectronics Reliability, 2009. – Vol. 49. – Iss. 12. – P. 1578-1585.
5. **Jusas V., Motiejūnas K.** Generation of Functional Delay Test with Multiple Input Transitions // Information Technology and Control, 2007. - Vol. 36. - No. 3. – P. 259-267.
6. **Yi J., Hayes J. P.** The Coupling Model for Function and Delay Faults // Journal of Electronic Testing: Theory and Applications, 2005. - Vol. 21. - No. 6. – P. 631–649.
7. **Bareiša E., Jusas V., Motiejūnas K., Šeinauskas R.** On the Enrichment of Static Functional Test // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 3(91). – P. 9–14.
8. **Ferrandi E., Ferrara G., Sciuto D., Fin A., Fummi F.** Functional Test Generation for Behaviorally Sequential Models // Proceedings of the Conference on Design, Automation and Test in Europe (DATE'01), 2001. – P. 403-410.
9. **Malaiya Y. K.** Antirandom testing: getting the most out of black-box testing // Proceedings of the 6th IEEE International Symposium on Software Reliability Engineering (ISSRE ’95). - Toulouse, France, October 1995. – P. 86–95.
10. **von Mayrhauser A., Chen T., Hajjar A., Bai A., Anderson C.** Fast antirandom (FAR) test generation // Proceedings of the 3rd IEEE International High-Assurance Systems Engineering Symposium (HASE ’98). - Washington, DC, USA, November 1998. – P. 262–269.
11. **Wu S. H., Jandhyala S., Malaiya Y. K., Jayasumana A. P.** Antirandom Testing: A Distance-Based Approach // VLSI Design, 2008. – Vol. 2008. DOI:10.1155/2008/165709.

Received 2010 06 30

E. Bareiša, P. Bieliauskas, V. Jusas, A. Targamadzė, L. Motiejūnas, R. Šeinauskas. Factor of Randomness in Functional Delay Fault Test Generation for Full Scan Circuits // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 9(105). – P. 39–42.

We investigated the influence of the random generation methods to the results of the functional delay test generation. There are the three possibilities: random generation of values 0 and 1 when the probability for the appearance of each value is 50% in every bit of test pattern; random generation of large integer values, which then are split into the sequences of 0’s and 1‘s according to the rules of the conversion of the decimal number to the binary number; anti-random generation when the presence of the earlier generated patterns is taken into account. We adopted the anti-random generation for the functional delay faults. The obtained results of the investigation indicate that the random generation has the direct influence to the results of the functional delay test generation. The largest coverage of transition fault is obtained when the anti-random generation is employed. Bibl. 11, tabl. 3 (in English; abstracts in English and Lithuanian).

E. Bareiša, P. Bieliauskas, V. Jusas, A. Targamadzė, L. Motiejūnas, R. Šeinauskas. Atsitiktinumo faktorius sudarant pilnutinio skleidimo nuosekliu schemų funkcinius vėlinimo gedimų tikrinimo testus // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2010. – Nr. 9(105). – P. 39–42.

Straipsnyje tyrinėjama atsitiktinių generavimo metodų įtaka funkcinių testų sudarymo kokybei. Galimi trys atsitiktinių testų generavimo variantai: 0 ir 1 generavimas, kai kiekvieno pasirodymo tikimybė yra 50 %; didelių sveikųjų skaičių, kurie vėliau išskaidomi į 0 ir 1, generavimas; antiatsitiktinis generavimas, įvertinant anksčiau buvusius rinkinius. Antiatsitiktinį generavimą pritaikėme funkcinių vėlinimo gedimų tikrinimo testams sudaryti. Pasiekti rezultatai rodo, kad atsitiktinio generavimo metodai turi tiesioginę įtaką sudarytų funkcinių testų kokybei. Didžiausia vėlinimo gedimų visuma buvo pasiekta antiatsitiktiniu generavimu. Bibl. 11, lent. 3 (anglų kalba; santraukos anglų ir lietuvių k.).