ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 *2011. No. 8(114)*
ELEKTRONIKA IR ELEKTROTECHNIKA

ELECTRONICS
T 170
ELEKTRONIKA

An Investigation of Possibilities of Improving Random Test Generation for Non-scan Sequential Circuits

E. Bareisa, V. Jusas, K. Motiejunas
Software Engineering Department, Kaunas University of Technology,
Studentu str. 50-406, Kaunas, Lithuania, e-mail: kestutis.motiejunas@ktu.lt
R. Seinauskas
Information Technology Development Institute, Kaunas University of Technology
Studentu str. 48a, Kaunas, Lithuania

crossref http://dx.doi.org/10.5755/j01.eee.114.8.685

Introduction

Semiconductor devices are becoming increasingly complex in terms of transistor count, frequency and integration. The distance between transistors is scaling down. The adoption of nanometer processes results in the new classes of defects that affect signal timing. The defect spectrum now includes more problems such as high impedance shorts, in-line resistance, and crosstalk between signals, which are not always detected with the traditional static-based tests, known as stuck-at tests. Detecting these delay defects requires a test approach that can apply test patterns at the rated speed of the device under test.

Test generation is being 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 technique of design for testability (DFT) is applied during the design of such circuits. This helps to reduce the cost of test development. But the scan design allows a synchronous sequential circuit to be brought into the states that the circuit cannot visit 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 disadvantage of DFT approach is that it adds extra delay to the circuit.

The other important direction of test generation is the functional test development at the 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, 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 functional test is very valuable also for the testing of intellectual property (IP) components whose implementation details are not known to the designer of the system on a chip.

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 task 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.

Random test sequences may be used for manufacturing testing as well as for simulation-based design verification [1–4]. The presented in [3, 4] research shows that functional tests designed using random test generation exhibit better transition fault coverages than tests produced by deterministic ATPG tools. Therefore, we are going to investigate further the possibilities of improving random test generation for at-speed testing of non-scan synchronous sequential circuits in this paper.

The rest of the paper is organized as follows. Section 2 presents the related work. In Section 3, we describe the

investigated approaches of random test generation and the experimental results. Section 4 concludes the paper.

Related works

The sequential circuit is comprised of two parts: the combinational logic and the flip-flops synchronized by a common clock signal. Only the primary inputs (PIs) of the circuit are controllable and the primary outputs (POs) are observable. For delay fault testing of non-scan sequential circuits, test application consists of the following steps: (a) initialization of the circuit to a known state, (b) fault activation to stimulate the fault being tested and (c) propagation of the fault effect to a primary output (PO). It may require a number of input vectors to initialize the circuit and to propagate the fault effects to a PO [5].

The application of random test sequences for testing of non-scan sequential circuits is presented in [1, 2]. It was shown in [1] that random primary input sequences achieve low fault coverage for synchronous sequential circuits due to the fact that they repeatedly assign the same values to subsets of state variables. To address this issue, in [1] a procedure is described for modifying a random primary input sequence to eliminate the appearance of input vectors that synchronize subsets of state variables. It is demonstrated that this procedure has a significant effect on the fault coverage that can be achieved by random primary input sequences.

However, the presented in [2] research shows that relatively long random test sequences exhibit better transition fault coverages than tests produced by deterministic ATPG tools. The paper [2] presents an approach for dividing of long test sequences into subsequences. The application of this approach allows increasing the fault coverage of the initial random generated test sequence and minimizing the length of the test by eliminating subsequences that don't detect new faults.

Under the approach presented in [6], the input vectors comprising the test sequence are fixed in advance. The process of generating the test sequence consists of ordering a set of precomputed input vectors such that the resulting test sequence has as high fault coverage as possible. The test generation process thus searches a limited set of input vectors for an appropriate order instead of exploring a search space that is limited only by the number of primary inputs of the circuit. However, only stuck-at faults are considered.

The delay fault test for non-scan synchronous sequential circuits could be constructed at the functional level using the software prototype model, as well [3, 4, 7].

The paper [3] presents an approach of test generation for non-scan synchronous sequential circuits based on functional delay models. The non-scan sequential circuit is represented as the iterative logic array model consisting of k copies of the combinational logic of the circuit. The value k defines a length of clock sequence. Especially, the functional delay test generation method is useful for the circuits, when the long test sequences are needed in order to detect transition faults. The paper [4] investigates the application of tests that are generated at functional level for detection of gate-level transition faults. Based on experimental results, there is developed a framework of test generation for non-scan sequential circuits. In [3, 4] provided comparison of experimental results demonstrate the superiority of the delay test patterns constructed at the functional level using the functional fault models against the transition test patterns generated at the gate level by deterministic test pattern generator.

Kang et al. [7] suggested the input/output transition (TRIO) fault model for functional test selection at the register-transfer level (RTL). It is defined with respect to the primary inputs, primary outputs, and state variable of the module. However, this model is approximate because it does not stipulate toggle propagation all the way to the primary outputs.

For synchronous sequential circuits, one important issue is their initialization, which means the sequential circuit must start from a known initial state for it to operate correctly, as well as when generating tests for circuit, or when verifying the functional properties [8]. In this paper we don't concern about initialization, i.e. we assume that the considered circuits have a *reset* line.

Further in the paper we will use terms "functional delay fault" and "subsequence" as defined in [4] and [2] respectively.

Definition 1. A functional delay fault is a tuple (I, O, tI, tO), where I is a primary input x_i (i=1, …, n) or a bit of previous state q_l (l=1, …, v) of the generic cell, O is a primary output y_j (j=1, …, m) or a bit of next state p_k (k=1, …, v), tI is a rising or falling transition at I, and tO is a rising or falling transition at O. [4]

Definition 2. The subsequence is a sequence of input patterns which starts with a set of initialisation patterns. The subsequence is composed of two parts of input patterns: the first part Sub(In) is a set of initialisation patterns that lead the circuit to the known state and the second part Sub(test) is a set of test patterns.[2]

The number of input patterns in Sub(test) defines the length of the subsequence.

Investigation of possibilities of improving random generation

At first, we are going to investigate the influence of distribution of "1" and "0" in randomly generated test pattern (number of "1" and "0" on average in randomly generated test pattern) to test quality. We generate randomly the whole particular subsequence S and then define using fault simulation the set of functional delay faults that S detects. If S detects any not yet detected faults, S is included into resulting test sequence.

The non-scan version of ITC'99 sequential synchronous benchmark circuits was chosen for experiments. We consider functional delay faults. The models of the benchmark circuits are written in C programming language. The lengths of subsequences of particular circuits are taken from [2].

The summarised results of experiments with various distributions of "1" and "0" are presented in Table 1. For each circuit, circuit name (Circuit), number of detected functional delay faults (Det. FD) and obtained sequence length (Length) are provided. Distribution of "1" is given in per cent. The notation "70%" (60%, 50%,… etc) means that there is on average 70% (60%, 50%,… etc) of "1" in each randomly generated test pattern.

Table 1. The influence of distribution of "1" to test quality

Circuit	70%		60%		50%		40%		30%	
	Det. FD	Length	Det. FD	Length	Det. FD	Length	Det. FD	Length	Det. FD	Length
b04	1797	4410	1797	3755	1797	3753	1797	3990	1795	4663
b06	89	85	89	64	89	75	89	73	89	96
b07	69	433	69	617	69	541	70	624	70	629
b08	206	1475	206	893	206	931	206	1383	206	1449
b09	526	2283	526	2393	526	2927	526	3569	526	4790
b10	292	654	292	627	292	654	291	830	290	1013
b11	957	38264	957	23979	957	36268	957	21742	956	28871
b12	272	14280	293	11376	295	7360	297	3719	310 637*	6595 32636*
b13	353	11517	352	12098	353	11906	353	11350	353	16941
Average	526	7736	529	5463	529	6564	529	4491	530	6013
b14	19739	353419	19244	362841	19120	364663	20184	367587	26613 33224**	405283 450066**
b15	16602	23102	14665	20487	13402	17489	12805	16735	10816	13486
Average b14, b15	18171	188261	16955	191664	16261	191076	16495	192161	18715	209385
Av. all	3718	40902	3499	39921	3373	40597	3416	39237	3820	43983

*Note: *- distribution 5%; ** - distribution 20%*

We made for each case two independent test generations in order to obtain more reliable experimental results. The distribution of '1" was taken between 30% - 70%. We present in Table 1 averaged outcome. For particular circuit the test generation times are the same for each distribution of "1". The quality of test specifies two factors – first the number of detected faults and second the test length. We assume that the best test is that, which detects the most faults. If there are two or more tests that detect the same number of faults, the best test is the shortest one. In Table 1, the best results are highlighted. Further, we assume that the test generation process saturates (all or almost all faults are already detected) when the total number of generated and analysed random subsequences exceeds the number of the last selected subsequence multiplied by a coefficient K.

Now let's analyse the results presented in Table 1. The test generation process saturates very fast for the circuits b04-b13. The minimal obtained value of K was 6, whereas it was shown in [4] that already the value 2 of K allows getting tests of satisfactory quality. If we compare the results of particular circuit (b04-b13), we can see that the test quality differs very little for various distributions of "1". Therefore, we made a conclusion that if test generation process saturates fast, then the distribution of "1" and "0" in randomly generated test pattern is not an important factor. Different circumstances are for the circuits b14 and b15. The biggest obtained value of K was 1.12, what means that the test generation process was far from saturation point. And in this case, we see that the distribution of "1" is an important factor for test quality. The number of detected functional delay faults differs significantly for various distributions of "1". For the circuit b14 the maximal number of detected faults is 26613 by distribution of "1" 30%, whereas the minimal number of detected faults is 19120 by distribution of "1" 50%. Similarly, the numbers of detected faults differ for the circuit b15 (maximal number 16602 by distribution of "1" 70% and minimal number 10816 by distribution of "1" 30%). Based on these sightings we suggest guidance for management of test generation process:

1. At the beginning perform many short test generations using various distributions of "1";
2. Find out the distribution of "1", using which the most faults are detected;
3. Generate functional delay test using chosen distribution of "1" until test generation process saturates.

Next, we implemented two semi-deterministic algorithms for construction of separate subsequences. We call them semi-deterministic because all test patterns are random; however, we generate randomly for every test pattern (except first one) in subsequence M candidates and include into subsequence that pattern which has the maximal weight. The test pattern weight is calculated as follows

$$W = N_{\text{det}} c1 + (N_{act} + N_{prop}) \cdot c2 , \qquad (1)$$

where N_{det} is the number of new (not yet detected on previous test subsequences) functional delay faults that are detected on considered test pattern; N_{act} is the number of new functional delay faults that are activated on considered test pattern; N_{prop} is the number of new functional delay faults that are propagated on considered test pattern; c1 and c2 are coefficients which values are chosen freely.

We present the Algorithm 1 and Algorithm 2 in Fig. 1 and Fig. 2 respectively, where p_{last} is the last already included into subsequence test pattern. The main difference between these two algorithms lies in that that Algorithm 1 stresses more the fault activation phase of test construction process, whereas Algorithm 2 emphasizes more the fault propagation phase.

For comparison we provide best results from Table 1 in columns 2-4, where the employed distribution of "1" is given in column 4 under heading "%".We made for each case two independent test generations as well and present

in Tables 2 and 3 the averaged outcome of application of Algorithms 1 and 2.

For i=1 to M do;
Generate randomly test pattern p_i
For p_{last} and p_i perform fault activation and determine N_{det} and N_{act};
For p_i perform fault propagation and determine N_{prop} and $N1_{det}$;
$N_{det}= N_{det} + N1_{det}$
Calculate the weight W_i of p_i
($W_i=N_{det}*c1 + (N_{act} + N_{prop})*c2$);
End for;
Include into subsequence test pattern p_i that has the biggest weight W_i;
Add newly detected faults to list of detected faults;
Refresh the list of faults that need propagation.

Fig. 1. Description of Algorithm 1

For i=1 to M do;
Generate randomly test pattern p_i
For p_i perform fault propagation and Determine N_{prop} and $N1_{det}$;
Calculate the weight W_i of p_i
($W_i=N1_{det}*c1 + N_{prop}*c2$);
End for;
Include into subsequence test pattern p_i that has the biggest weight W_i;
For p_{last} and chosen p_i perform fault activation;
Add newly detected faults to list of detected faults;
Refresh the list of faults that need propagation.

Fig. 2. Description of Algorithm 2

The experimental results of application of both algorithms for functional delay test generation are presented in Table 2 and Table 3.

Table 2. Results of application of Algorithm 1

Circuit	Best of Table 1			Algorithm 1						
				M	c1=100; c2=1		c1=1; c2=100		c1=1; c2=1	
	Det. FD	Length	%		Det. FD	Length	Det. FD	Length	Det. FD	Length
b04	1797	3753	50%	10	1797	3788	1797	3584	1797	3790
b06	89	64	60%	7	89	81	89	75	89	83
b07	70	624	40%	3	69	538	69.5	605	69.5	569
b08	206	893	60%	10	206	825	206	1151	206	1164
b09	526	2283	70%	3	526	3240	526	3098	526	2946
b10	292	627	60%	10	292	616	292	647	291.5	635
b11	957	21742	40%	10	957	38153	957	47316	957	41195
b12	637	32636	5%	10	608	28957	601	25018	605	27518
b13	353	11350	40%	10	353	12561	353	15396	353	12554
Average	547	8219	-	-	544	9862	543	10766	544	10050
b14	33224	450066	20%	30	28900	150951	32909	428715	27944	119628
b15	16602	23102	70%	30	15917	22502	15949	22566	15825	22951
Average b14, b15	24913	236584	-	-	22408.5	86726.5	24429	225640.5	21884.5	71289.5
Av. all	4978	49740	-	-	4519	23837	4886	49834	4424	21185

Table 3. Results of application of Algorithm 2

Circuit	Best of Table 1			Algorithm 2						
				M	c1=100; c2=1		c1=1; c2=100		c1=1; c2=1	
	Det. FD	Length	%		Det. FD	Length	Det. FD	Length	Det. FD	Length
b04	1797	3753	50%	10	1797	3807	1797	3715	1797	3645
b06	89	64	60%	7	89	77	89	60	89	76
b07	70	624	40%	3	70	742	69	693	69	490
b08	206	893	60%	10	206	1179	206	1156	206	1407
b09	526	2283	70%	3	526	3020	526	2973	526	2863
b10	292	627	60%	10	292	658	291.5	667	291.5	630
b11	957	21742	40%	10	957	40154	957	42777	957	36478
b12	637	32636	5%	10	641	36452	643	35917	641	34949
b13	353	11350	40%	10	353	13215	353	15396	353	16929
Average	547	8219	-	-	548	11034	548	11484	548	10830
b14	33224	450066	20%	30	32525	385010	32909	428715	32763	413520
b15	16602	23102	70%	30	15248	23317	15761	23463	16507	21902
Average b14, b15	24913	236584	-	-	23886.5	204163.5	24335	226089	24635	217711
Av. all	4978	49740	-	-	4791	46148	4873	50503	4927	48444

Both algorithms used the presented in column 4 distributions of "1". The test generation took for particular circuit the same time as for obtaining presented in Table 1 results. The highest quality tests are highlighted.

We applied Algorithms 1 and 2 using various values of coefficients c1 and c2 trying to take more stress on detecting new faults (c1=100) or taking more stress on the propagation of faults (c2=100); and the third variant was when both coefficients are equal. We related great expectations with implementation of algorithms of semi deterministic construction of test subsequences. We expected that in case when the test generation process doesn't saturate we will get higher or even much higher fault coverages; and in case of fast saturation of the test generation process we anticipated that the obtained test will be shorter. However, our expectations don't materialize at all. If we compare the semi deterministic test generation results with pure random search, we see that the pure random search produced best results for 6 circuits (total number of considered circuits was 11). The averaged results display the superiority of the pure random search as well. There may be a doubt that probably the values of M were too little. However, the provided results for circuits b07, b09 (both have only 1 primary input) and b06 (2 primary inputs) for which the exhaustive search of candidates was used deny this assumption. Therefore, we made a conclusion: the optimisation of separate steps by construction of test subsequences doesn't improve the final outcome.

Conclusions

Sequential circuit testing has been recognized as the most difficult problem in the area of fault detection. High-performance circuits with aggressive timing constraints are usually very susceptible to delay faults. The latest research shows that functional tests designed using random test generation exhibit good transition fault coverages. Therefore, we investigated further the possibilities of improving random test generation for at-speed testing of non-scan synchronous sequential circuits. Based on research of distribution of "1" in randomly generated test pattern we suggested guidance for management of test generation process. The implementation of semi deterministic algorithms showed that the optimisation of separate steps by construction of test subsequences doesn't improve the final outcome.

References

1. **Pomeranz I., Reddy S. M.** Primary input vectors to avoid in random test sequences for synchronous sequential circuits // IEEE transactions on computer–aided design of integrated circuits and systems, 2008. – No. 27(1). – P. 193–197.
2. **Bareiša E., Jusas V., Motiejūnas K., Šeinauskas R.** The non–scan delay test enrichment based on random generated long test sequences // Information Technology and Control, 2010. – No. 39(4). – P. 251–256.
3. **Bareiša E., Jusas V., Motiejūnas L., Šeinauskas R.** Generating Functional Delay Fault Tests for Non–Scan Sequential Circuits // Information Technology and Control, 2010. – No. 39(2). – P. 100–107.
4. **Bareiša E., Jusas V., Motiejūnas K., Šeinauskas R.** On delay test generation for non–scan sequential circuits at functional level // Electronics and Electrical Engineering. – Kaunas: Technologija, 2011. – No. 3(109). – P. 67–70.
5. **Pant P., Chatterjee A.** Path–delay fault diagnosis in non–scan sequential circuits with at–speed test application // Proceedings of the International Test Conference (ITC'2000), 2000. – P. 245–252.
6. **Pomeranz I., Reddy S. M.** TOV: sequential test generation by ordering of test vectors // IEEE transactions on computer–aided design of integrated circuits and systems, 2010. – No. 29(3). – P. 454–465.
7. **Kang J., Seth S. C., Gangaram V.** Efficient RTL Coverage Metric for Functional Test Selection // Proceedings of the 25th IEEE VLSI Test Symposium, 2007. – P. 318–324.
8. **Morkūnas K., Šeinauskas R.** Circuit Reset Sequences based on Software Prototypes // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 7(103). – P. 71–76.

Received 2011 04 18

E. Bareisa, V. Jusas, K. Motiejunas, R. Seinauskas. An Investigation of Possibilities of Improving Random Test Generation for Non-scan Sequential Circuits // Electronics and Electrical Engineering. – Kaunas: Technologija, 2011. – No. 8(114). – P. 11–15.

High-performance circuits with aggressive timing constraints are usually very susceptible to delay faults. The latest research shows that functional tests designed using random test generation exhibit good transition fault coverages. In the paper, we investigated the possibilities of improving random test generation for at-speed testing of non-scan synchronous sequential circuits. Based on research of distribution of "1" in randomly generated test pattern we suggested guidance for management of test generation process. The implementation of semi deterministic algorithms showed that the optimisation of separate steps by construction of test subsequences doesn't improve the final outcome. Ill. 2, bibl. 8, tabl. 3 (in English; abstracts in English and Lithuanian).

E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas. Nuoseklių schemų atsitiktinio testų generavimo pagerinimo galimybių tyrimas // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2011. – Nr. 8(114). – P. 11–15.

Šiuolaikinės schemos yra labai jautrios vėlinimo gedimams. Naujausi tyrimai rodo, kad funkciniai testai, sukonstruoti remiantis atsitiktiniu testinių rinkinių generavimu, gerai aptinka perėjimo gedimus. Šiame straipsnyje nagrinėjamos atsitiktinio testų generavimo pagerinimo galimybės. Remiantis tyrimais, atliktais su skirtingu vienetukų ir nuliukų pasiskirstymu atsitiktinai sugeneruotame rinkinyje, pasiūlyta testų generavimo proceso valdymo metodika. Įdiegus nevisiškai deterministinius algoritmus, pasirodė, kad, esant tai pačiai skaičiavimo apimčiai atsitiktinės paieškos metu dalinių žingsnių optimizavimas nepagerina galutinio rezultato. Il. 2, bibl. 8, lent. 3 (anglų kalba; santraukos anglų ir lietuvių k.).