On Test Generation with A Limited Number of Tests

Hideyuki Ichihara†, Seiji Kajihara‡†, Kozo Kinoshita†

† Dept. of Applied Physics, Osaka University
‡ Computer Science and Electronics Dept., Kyushu Institute of Technology

Abstract

This paper considers a new test generation scheme in which a limitation of the number of tests exists. Since, in this scheme, correct fault coverage cannot be calculated by the representative faults, we present a method for calculating the correct fault coverage by using the weighted fault list. And then we propose a selection-based test generation method which derives limited number of tests with higher fault coverage. The experimental results for IDDQ testing shows that our test generation method can generate tests with fault coverage close to the maximum fault coverage.

1. Introduction

Many test generation algorithms have been proposed [1]-[8]. These algorithms were developed so as to satisfy the following requirements.

1. practical computation time,
2. high fault coverage (fault efficiency),
3. small set of generated tests.

Recently, researches of test generation have focused on generating a compact test set [3]-[8] to reduce test application time and test storage requirement of LSI testers. Especially for test generation for IDDQ teasing [5]-[8], test compaction is strongly desired because the measurement of IDDQ is time-consuming. In spite of the strong requirement of test compaction, the most researches on test compaction tend to consider that fault efficiency is more important than the number of generated tests, i.e. the objective is to generate a smaller test set without losing fault efficiency. Therefore, even if the minimum test set could be generated, the size of tests may be too large to apply all the tests.

In this paper, we present a new test generation scheme under the condition that there exists an upper limit of the number of tests. Namely, the fault coverage of a test set is maximized without exceeding the upper limit of the number of tests. Our purpose is to propose a test generation method in this new scheme. Before the proposal of our test generation method, we point out a problem in calculating fault coverage which occurs only on the new test generation scheme. Our solution of this problem is the use of the weighted fault list. Next, we propose a test generation method to generate tests with higher fault coverage, and give experimental results by applying the proposed method to IDDQ test generation.

The remainder of this paper is organized as follows. In Section 2, we explain the problem of the fault coverage and the method to calculate the correct fault coverage. In Section 3, we propose a test generation method when the number of tests has an upper limit. In Section 4, experimental results for benchmark circuits are shown. Section 5 is the conclusion of this paper.

2. Whole fault coverage for representative faults

In general, a fault model, which abstracts the behavior of physical defects, is introduced into test generation and evaluation of a generated test set. For example, stuck-at fault, delay fault, and bridging fault are well-known fault models. For a fault model, a fault list for a circuit under test is constructed by fault collapsing. Fig. 1 shows an image of fault collapsing. If fault f_a can be detected by any test to detect fault f_b, it is not necessary to contain f_a in the fault set. The fault collapsing contributes to the reduction of computation time of test generation and test evaluation. Faults selected by the fault collapsing are called *representative faults*.

In conventional test generation, the number of tests

0-7695-0104-4/99 $10.00 © 1999 IEEE

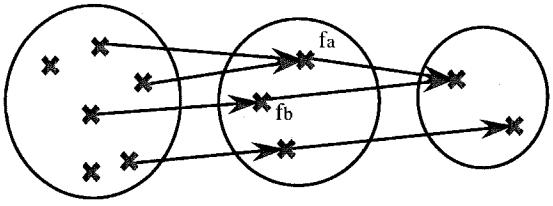

Figure 1. Fault modeling and fault collapsing

is not limited, and representative faults are used for the calculation of fault coverage. We call the fault coverage *representative fault coverage*. On the other hand, the fault coverage for all faults is called *whole fault coverage*. Generally the representative fault coverage is different from the whole fault coverage. If tests for all detectable faults can be generated, a test set with the highest representative fault coverage has the highest whole fault coverage. However, if the number of tests is limited, this is not always true.

An example is shown for stuck-at faults of 2-inputs AND gate of Fig. 2. Table 1 and Table 2 are fault tables for the all faults and the representative faults, respectively. Namely, the former table can derive the whole fault coverage and the latter table the representative fault coverage. In these tables, fault a-0 denotes stuck-at-0 fault on signal line a. Suppose that the number of tests to be generated is limited to 1, i.e. a test which can detect the most faults of four tests $(b,c) = \{(0,0),(0,1),(1,0),(1,1)\}$ is selected. As can be seen in Table 1, in the case where a test is generated by using whole fault coverage, test $(1,1)$ is selected. As can be seen in Table 2, in the case where a test is generated by using the representative fault coverage, test $(0,1)$ or $(1,0)$ is selected. Thus, it is recognized that tests generated by using the representative fault coverage are different from those by using the whole fault

Table 1. Fault table of AND gate

fault	test (b,c)			
	(0, 0)	(0, 1)	(1, 0)	(1, 1)
a-0				○
a-1	○	○	○	
b-0				○
b-1		○		
c-0				○
c-1			○	

Figure 2. AND gate

Table 2. Representative fault table

fault	test (b,c)			
	(0, 0)	(0, 1)	(1, 0)	(1, 1)
a-0				○
a-1	○	○	○	
b-1		○		
c-1			○	

coverage. Since tests with high whole fault coverage may be able to detects more physical defects than that with high representative fault coverage, it is desired to generate tests with high whole fault coverage.

The above example means that the representative fault coverage should not be used in test generation if the number of tests is limited, although representative faults contribute to the reduction of computation efforts. In order to calculate whole fault coverage using representative faults, we propose a *weighted fault list*. Each representative fault in this list has a weight which denotes the number of faults represented by the representative fault. Table 3 shows a weighted fault list for representative faults shown in Table 2. For example, since the weight of representative fault a-0 is 3, a test for detecting a-0 detects three other faults which are not representative faults. Using the weighted fault list, whole fault coverage can be calculated.

3. Test generation method under a limitation of the number of tests

In this section, we propose a test generation method for generating n tests with the *higher* fault coverage,

Table 3. Weighted fault list

representative fault	weight
a-0	3
a-1	1
b-1	1
c-1	1

```
1  Selection-Based_TG(n, T_g)
2     n : a number of tests to be selected
3     T_g : a given test set
4  {
5     Ts: set of selected tests
6     WFL : weighted fault list

7     Set T_s = φ
8     Construct WFL;
9     While(|T_s| ≤ n){
10       Calculate the whole fault coverage of each test
             in T_g on WFL;
11       Select test t with the highest whole fault
             coverage from T_g;
12       Move t from T_g to T_s;
13       Remove faults which are detected by t from
             WFL;
14    }
15    return T_s;
16 }
```

Figure 3. Algorithm of selection-based TG

where n is the given number of tests. A simple method for this objective is to investigate the fault coverage of every combination of n tests. The number of combinations of n tests is ${}_{2^p}C_n = 2^p!/((2^p - n)!n!)$, where p is the number of primary inputs. If p becomes large and n becomes close to 2^{p-1}, it is not preferable to calculate fault coverage of all combinations. Therefore, our method selects n tests from a given set of m tests ($n < m \leq 2^p$). The given test set is arbitrary, hence the proposed method does not depend on the given set. We typically use a test set with maximum fault coverage. n tests are greedily selected step by step. Fig. 3 shows a procedure of our method. This procedure receives the number of tests and a set of tests which is a target for selecting. First, the weighted fault list is constructed (line 8). Next, the whole fault coverage of every test on the weighted fault list is calculated (line 10), and the test with the highest whole fault coverage is selected from a given test set (line 11). The selected test is stored and faults detected by the test are removed from the weighted fault list (line 12, 13). While the number of selected tests is less than the number of tests to be selected, test selection process is continued. Finally, this algorithm returns a selected test set.

4. Experimental results

We applied the proposed test generation method to IDDQ testing for bridging faults. In order to choose representative faults for bridging faults, we also propose a procedure based on the functional equivalence of signal lines using implication relation analysis. The representative faults derived by the proposed procedure are fewer than that by the topological analysis [8], because our procedure finds greater number of pairs of equivalent signal lines than the topological analysis.

Table 4. Difference of calculation methods of fault coverage

	whole fault coverage	
circuit	weighted list	non-weighted list
c880	92.37	92.39
c1355	84.00	84.00
c1908	88.44	88.72
c2670	91.56	91.40
c3540	90.86	90.75
c5315	90.89	90.77
c7552	91.76	91.16

(n=5)

Table 4 shows a result for ISCAS'85 benchmark circuits where 5 tests were selected. The given set T_g is a set of tests for stuck-at faults generated by the test generation method based on SOCRATES [2]. Since the test set for stuck-at faults can set 0 and 1 to all signal lines once at least, it is preferable to a test set to be targeted for our method. The column "weighted list" in Table 4 is the whole fault coverage of the test set selected by our method. In order to explain the difference between test sets generated by whole fault coverage and by representative fault coverage, we perform our proposed method using the representative fault coverage. Namely, instead of the weighted fault list, a non-weighted fault list containing only representative faults is constructed. The column "non-weighted list" in Table 4 is the whole fault coverage of the selected tests. For c7552, the whole fault coverage of tests selected by using the weighted fault list is 91.76%, which is greater than that by using the non-weighted fault list, i.e. 91.16%. For 5 circuits out of 7, the whole fault coverage using the weighted fault list is greater than those without using the weighted fault list.

Table 5 shows the whole fault coverage of selected tests whose number is a lower bound of the minimum number needed to detect all faults in each circuit. The column "#node" means the number of signal lines, except fanout branches, of each circuits. The column "#test" means the number of selected tests, which is a lower bound of tests to detect all faults. It is calculated by $\lceil \log_2(N) \rceil$, where N is the number of nodes shown

Table 5. Experimental Result of selection-based TG

circuit	#node	#test	whole fault coverage
c432	198	8	99.3
c499	245	8	96.1
c880	445	9	99.5
c1355	589	10	96.9
c1908	911	10	99.1
c2670	1428	11	99.7
c3540	1721	11	99.2
c5315	2487	12	99.8
c6288	2450	12	99.8
c7552	3712	12	99.8

Table 6. Experimental results for selection of tests with lower fault coverage

circuit	#test	whole fault coverage
c432	8	75.8
c499	8	75.7
c880	9	95.2
c1355	10	91.7
c1908	10	96.9
c2670	11	97.2
c3540	11	95.2
c5315	12	98.8
c6288	12	97.0
c7552	12	94.2

in the second column in Table 5. For c432, at least 8 tests are needed to detect all bridging faults. The column "whole fault coverage" shows the whole fault coverage of selected tests. Our method can select a test set with over 99% fault coverage, i.e. close to 100%, for circuits except c499 and c1355, although the selection strategy of our method is greedy.

To verify the effectiveness of our greedy selection, we implemented an algorithm to select tests with lower fault coverage. Namely, the process at line 11 in Fig. 3 is replaced by selecting a test with the lowest fault coverage from a given test set. The fault coverage of test set which is selected by the method are shown in Table 6. The fault coverage shown in Table 6 is naturally less than those shown in Table 5, however the difference of whole fault coverage of two methods is comparatively large. For c432, the difference of whole fault coverage of two methods is no less than 24.5%. We see from these results that our selection can derive better tests with higher fault coverage.

5. Conclusion

We considered a new test generation scheme which the number of tests has an upper bound. In this scheme, the fault coverage calculated by representative faults is not accurate. Therefore we proposed the calculation of the fault coverage using weighted fault list. Our proposed test generation method was based on selection of limited number of tests. Experimental results showed the effectiveness of our method. In the future work, we will apply this test generation scheme to IDDQ testing for sequential circuits.

References

[1] M. Abramovici, M. A. Breuer and A. D. Friedman, *Digital Systems Testing and Testable Design*, Computer Science Press, 1990.

[2] M. H. Schulz, E. Trischler and T. M. Sarfert, "SOCRATES: A Highly Efficient Automatic Test Pattern Generation System," *IEEE Trans. Computer-Aided Design*, pp. 126-137, 1988.

[3] P. Goel and B. C. Rosales, "Test Generation and Dynamic Compaction of Tests," Test Conf., pp. 189-192, Oct. 1979.

[4] I. Pomeranz, L. N. Reddy and S. M. Reddy, "COMPACTEST: A method to generate compact test sets for combinational circuits," *IEEE Trans. Computer-Aided Design*, pp. 1040-1049, 1993.

[5] W. Mao, R. K. Gulati, D. K. Goel and M. D. Ciletti, "QUIETEST: A Quiescent Current Testing Methodology for Detecting Leakage Faults," Int'l Conf. on CAD, pp.280-283, 1990.

[6] S. Chakravarty and P. J. Thadikaran, "Algorithms to Select IDDQ Measurement Points to Detect Bridging Faults," *Journal of Electronic Testing: Theory and Application*, Vol. 8, pp.275-285, 1996.

[7] T. Shinogi and T. Hayashi, "A Simple and Efficient Method for Generating Compact IDDQ Test Set for Bridging Faults," VLSI Test Sympo., pp.112-117,1998.

[8] R. S. Reddy, I. Pomeranz, S. M. Reddy and S. Kajihara, "Compact Test Generation for Bridging Faults under IDDQ testing," VLSI test Sympo., pp.310-316, 1995.