Scalable Antirandom Testing (SAT)

Muhammad Sadiq Sahari, Abu Khari A'ain, Ian A. Grout

***Abstract*— Antirandom testing approach requires large input space and complex test vector generation algorithm when used on circuit under test (CUT) with large number of inputs. In this work, we proposed a novel and simple approach of Antirandom sequence generation by using the least significant bit (LSB) of the test vector as a reference to generate the next test vector. Fault simulations on ISCAS'85 benchmark circuits shown that a high fault coverage for combinational logic circuits has been obtained. Another attractive feature of the proposed technique is the scalable of the algorithm that can be generate test vectors in short time even for CUT with large number of inputs.**

***Index Terms*— Integrated Circuit (IC) Testing, Built-in self-test (BIST), Test Pattern Generation (TPG), Pseudorandom Testing and Antirandom Testing.**

I. INTRODUCTION

In random test vector generation method, the generated test vectors are purely arranged in random, while in Antirandom (AT) method [1], the vectors are systematically arranged. Random testing [2] concept does not consider the previous (history of) test vectors which have been used in the production of the next test vectors. This results in test vectors capturing the same faults as the previous vectors and produces low [1] accumulated fault coverage (FC). AT testing was introduced to overcome this problem. It considers the previous test vectors when generating next test vectors by employing maximum Hamming distance and Cartesian distance as distance measurements between each test vectors which results in a high FC. The main problem with AT approach is it dependence on enumeration of the input space and computation of each input vector when used on an arbitrary set of existing test data. For example, if there is an n bit inputs and the test vectors that have been generated is m, then the search space for the next test vector will be $2^{(n-m)}$. Then the search space for the next test vector will be $2^{(n-m-1)}$. It can be seen that for large number of input bits, the search space is also large. Some enhancements of this method were introduced in [3-9], but they still need large search space.

Manuscript received om March 2015.
Muhammad Sadiq Sahari, PhD Student, Department of Microelectronics and Computer Engineering, Faculty of Electrical Engineering, Universiti Teknologi Malaysia.
Abu Khari A'ain, Prof., Department of Microelectronics and Computer Engineering, Faculty of Electrical Engineering, Universiti Teknologi Malaysia.
Ian A. Grout, Prof., Department of Electronics and Computer Engineering, University of Limerick.

Our objective is to find a less computational complexity approach to generate AT sequences that can easily generate large test vectors for large inputs CUT. Thus, we introduced a novel technique called Scalable Antirandom testing (SAT). The key idea of the proposed technique is using the least significant bit (LSB) of the test vector as a reference to generate the next test vector instead of using the distance between two test vectors like the normal AT testing.

II. ARCHITECTURE DESCRIPTION

Architecture description: In this paper, we proposed a novel approach in order to solve the problem of large computation space in AT. We used the least significant bit (LSB) of the test vector as a reference and swap the value of the LSB, $t_0(i)$ and its inverse, $!t_0(i)$ for every 2^n cycle as shown in equation (1).

$$t_n(i+1) = t_0(i) \textit{ and } !t_0(i) \textit{ swap for every } 2^n \textit{ cycle} \quad (1)$$

Where $n \geq 0$.

We generate one test vector at a time in the Scalable Antirandom based on the value of LSB as a reference. Let us illustrate with an example how to generate 4 bit Scalable Antirandom sequences:

1st (LSB):
$t_0(i+1) = t_0(i) \textit{ and } !t_0(i) \textit{ swap for every } 2^0 \textit{ cycle}$
$t_0(i+1) = \, !t_0(i)$

2nd bit:
$t_1(i+1) = t_0(i) \textit{ and } !t_0(i) \textit{ swap for every } 2^1 \textit{ cycles}$

3rd bit:
$t_2(i+1) = t_0(i) \textit{ and } !t_0(i) \textit{ swap for every } 2^2 \textit{ cycles}$

4th bit:
$t_3(i+1) = t_0(i) \textit{ and } !t_0(i) \textit{ swap for every } 2^3 \textit{ cycles}$

The complete sequences of the 4 bit SAT is shown on the Table 1. As can be seen from Table 1, the test vectors are arranged according to equation (1) which does not require search space to generate next test vectors. The method is easily scale up to generate large number of test vectors for large inputs CUT.

Published By:
Blue Eyes Intelligence Engineering & Sciences Publication

International Journal of Innovative Science and Modern Engineering
IJISME
www.ijisme.org
Exploring Innovation

Table 1: Generation of 4-bit Scalable Antirandom.Testing

Cycle	↓ t_0 (LSB)	↓ t_1	↓ t_2	↓ t_3
0	0	0	0	0
1	1	1	1	1
2	0	1	0	0
3	1	0	1	1
4	0	0	1	0
5	1	1	0	1
6	0	1	1	0
7	1	0	0	1
8	0	0	0	1
9	1	1	1	0
10	0	1	0	1
11	1	0	1	0
12	0	0	1	1
13	1	1	0	0
14	0	1	1	1
15	1	0	0	0

III. SIMULATION RESULTS

ISCAS'86 benchmark circuits have been employed as circuit under tests (CUTs) in order to evaluate the effectiveness of the SAT method. Table 2 and Fig. 1 show the comparison of FC between the SAT and AT pattern generation using c499 circuit. It shows that the SAT method obtains higher FC with significantly fewer test patterns compared to AT

Table 2: Comparison of FC between SAT and AT using c499 circuit

Test Pattern	1	5	10	50	75	100	200
SAT	*12.7*	*61.4*	*75.2*	*91.4*	*93.4*	*95.8*	*97.6*
AT	*12.7*	*49.3*	*69.9*	*88.1*	*92.4*	*93.4*	*97.2*

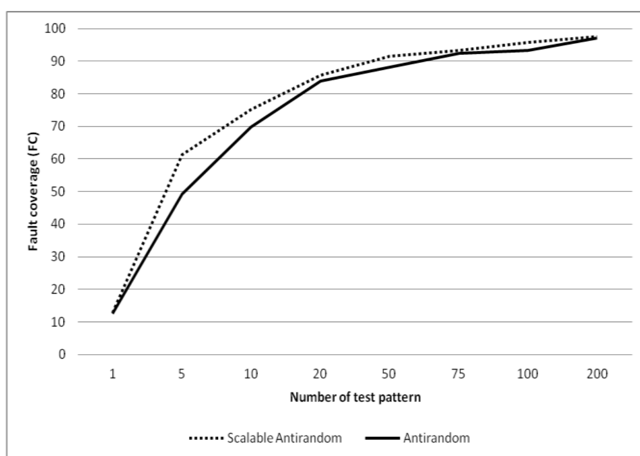

Fig. 1 Comparison of FC between Scalable Antirandom and Antirandom using c499 circuit

Table 3 presents the comparison of fault coverage between AT method and SAT using c1355, c880 and c3540 circuits. Both methods use initial seed value equal to 0 (all zeroes). Table 3 also indicates that the SAT method provides higher FC even when the test is sampled at early clock cycle.

Table 3: Comparison of FC between AT and SAT using c1355, c880 and c3540 circuits.

Test Vector	c1355		c880		c3540	
	AT	SAT	AT	SAT	AT	SAT
1	8.13	8.13	8.91	8.91	14.2	14.2
2	34.2	45.3	43.0	42.5	35.0	37.9
5	53.6	60.1	67.5	69.5	47.8	46.3
10	72.2	77.3	76.9	78.2	63.9	64.1
50	80.6	84.4	85.6	84.6	75.3	75.2
75	87.0	87.4	88.6	88.3	80.1	81.0
100	88.4	89.5	90.9	91.5	83.5	83.1
200	92.6	92.3	95.0	96.2	87.5	87.7

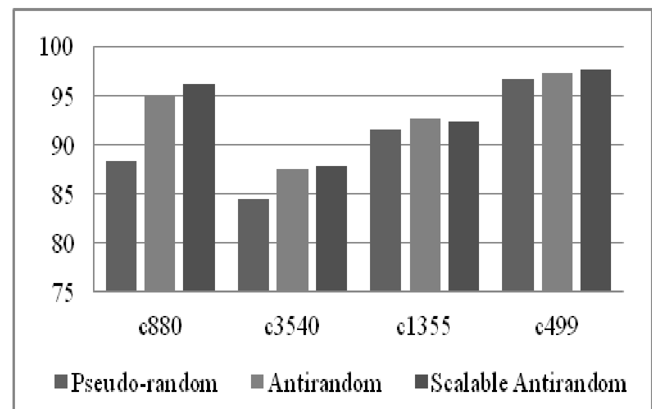

Fig. 2 Comparison of FC between Pseudorandom, AT and SAT using 200 test patterns.

Fig. 2 shows the comparison of fault coverage between Pseudo-random method, AT method [1] and SAT method. The graph presents the fault coverage obtained after tested with 200 test patterns. The SAT method gives the highest fault coverage: c880 (96.18%), c3540 (87.72%) and c499 (97.63%).

IV. CONCLUSIONS

The SAT method is proposed to eliminate the need for large space search and long computational time in AT for generation of large test vectors in large inputs CUT. Experimental results proved that the proposed technique can also produce high fault coverage using less test vectors. Another attractive feature of the proposed TPG method is the algorithm is scalable that can be easily employed for any number of inputs CUT.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

ACKNOWLEDGMENTS

We thank Virginia Polytechnic Institute & State University for providing ATALANTA, FSIM fault simulator and HOPE fault simulator to support this work.

REFERENCES

1. Wu, S.H., Jandhyala, S., Malaiya, Y.K., and Jayasumana, A.P.: ‘Antirandom testing: a distance-based approach’, VLSI Design, 2008, 2008, (2), pp. 2
2. Wagner, K.D., Chin, C.K., and McCluskey, E.J.: ‘Pseudorandom testing’, Computers, IEEE Transactions on, 1987, 100, (3), pp. 332-343
3. Mrozek, I., and Yarmolik, V.N.: ‘Iterative Antirandom Testing’, Journal of Electronic Testing, 2012, pp. 1-15
4. Singh, K., Kumar, R., and Kaur, I.: ‘Effective Test Case Generation Using Antirandom Software Testing’, International Journal of Engineering Science, 2010, 2
5. Tappenden, A.F., and Miller, J.: ‘A novel evolutionary approach for adaptive random testing’, Reliability, IEEE Transactions on, 2009, 58, (4), pp. 619-633
6. Chen, T., Bai, A., Hajjar, A., Andrews, A.K.A., and Anderson, C.: ‘Fast anti-random (FAR) test generation to improve the quality of behavioral model verification’, Journal of Electronic Testing, 2002, 18, (6), pp. 583-594
7. Zhou, Z.Q., Sinaga, A., and Susilo, W.: ‘On the fault-detection capabilities of adaptive random test case prioritization: Case studies with large test suites’, in Editor (Ed.)^(Eds.): ‘Book On the fault-detection capabilities of adaptive random test case prioritization: Case studies with large test suites’ (IEEE, 2012, edn.), pp. 5584-5593
8. Yiunn, D.B.Y., Bin A'ain, A., and Ghee, J.: ‘Scalable test pattern generation (STPG)’, in Editor (Ed.)^(Eds.): ‘Book Scalable test pattern generation (STPG)’ (IEEE, 2010, edn.), pp. 433-435
9. Shiyi, X., and Jianwen, C.: ‘Maximum distance testing’, in Editor (Ed.)^(Eds.): ‘Book Maximum distance testing’ (2002, edn.), pp. 15-20