Proceedings of the Fifth International Conference on Trends in Electronics and Informatics (ICOEI).
IEEE Xplore Part Number:CFP21J32-ART; ISBN:978-1-6654-1571-2

Automatic Test Pattern Generation of Multiple stuck-at faults using Test Patterns of Single stuck-at faults

Charitha Reddy P[1], SaiTulasi K[2], Anuja T J[3], Rajarajeswari R[4], Navya Mohan[5]
Department of Electronics and Communication Engineering,
Amrita School of Engineering, Coimbatore,
Amrita Vishwa Vidyapeetham,
India
[1]cb.en.u4ece16237@cb.students.amrita.edu, [2]cb.en.u4ece16224@cb.students.amrita.edu,
[3]cb.en.u4ece16210@cb.students.amrita.edu, [4]cb.en.u4ece16242@cb.students.amrita.edu,
[5]m_navya@cb.amrita.edu

2021 5th International Conference on Trends in Electronics and Informatics (ICOEI) | 978-1-6654-1571-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICOEI51242.2021.9452949

***Abstract*— The fabricated circuitries are getting massive and denser with every passing year due to which a normal automatic test pattern generation technique to detect only the single stuck-at faults will overlook the multiple stuck-at faults. But generating test patterns that can detect all possible multiple stuck-at fault is practically not possible. Hence, this paper proposes a method, where multiple faults can be detected by using test vectors for detecting single stuck-at faults. Here, the patterns for detecting single faults are generated and their ability to detect multiple stuck-at faults is also analyzed. From the experimental results it was observed that, the generated vectors for single faults cover maximum number of the multiple faults and then new test vectors are generated for the undetermined faults. The generated vectors are optimized for the compact test patterns in order to reduce the test power.**

Keywords*— *Automatic test pattern generation. Double stuck-at faults. Single stuck-at faults. Multiple stuck-at faults.

I. INTRODUCTION

After the fabrication process, it is crucial that the Integrated Circuits are tested before the product delivery to ensure that it functions correctly without any faults. There are many common fault models that determine the faults such as bridge faults, delay faults, stuck-at faults. Most common fault model which covers majority of the physical defects occurring in a circuit is stuck-at fault. Earlier approaches detected the single stuck-at faults that were embedded in the circuits and multiple faults were likely to be overlooked. The modern VLSI technology has allowed the semiconductor industry to make denser chips with a greater number of transistors in which multiple faults are inevitable and are more in number when compared to single faults. Now, detecting multiple faults in huge circuits would be a difficult task. Therefore, to reduce the complexity of the task single faults test sets are considered to determine the multiple faults.

Majority of the multiple stuck-at faults (MSA) can be predicted using single stuck-at fault (SSA) vectors. ATPG techniques are used to generate vectors for the MSA faults which cannot be detected earlier. Automatic test pattern generation is an efficient technology which can generate quick and compact test patterns. In this paper, Double stuck-at faults (DSA) are detected using the single stuck-at faults test patterns and this is extended to the Triple Stuck-at faults (TSA) and Quadruple Stuck-at faults(QSA), as practically the probability of more than four faults occurring in a circuit at the same time is very low [2].

The aim of the paper is to detect the double stuck-at faults using the test sets of single faults and then extend this implementation to TSA and QSA faults. This method will reduce the complexity in detecting the multiple faults.

The paper is organized as follows: Section 2 is brief description about the previous works done in this field, Section 3 illustrates the proposed methodology, Section 4 lists out the details of experimental results and analysis and finally section 5 is the conclusion and future scope of this work.

II. RELATED WORK

As the chips are getting denser due to the advancements made in the modern technology, faults are inevitable in the fabricated circuitries. Therefore, patterns that are generated to test the functionalities of the circuits are called test vectors. There are various fault models out of which stuck-at faults are the most common faults that are likely to occur [4][6]. ATPG technique has been used to create the compact test sets. Initially there were methods like D-algorithms, path-oriented decision making (PODEM) to determine the stuck-at faults. In

Authorized licensed use limited to: Rochester Institute of Technology. Downloaded on February 01,2023 at 16:54:03 UTC from IEEE Xplore. Restrictions apply.

the modern circuitry industry (SAT) based ATPG techniques were developed to determine the stuck-at faults even in the larger circuitries. This is an efficient way of generating compacted test patterns. [1] Since the number of transistors in the chips are augmenting and resulting in larger and complex circuits, multiple stuck-at faults (MSA) are difficult to check because of its enormous fault possibilities [2][5]. The whole MSA fault possibilities will be 3^q-1when q is the number of interconnections in the circuit. It is impossible to consider all possible cases individually when q is large. [1]

Earlier approach has proposed a method called parallel vector pair which is used to generate the test vectors for MSA. But these methods might take long processing time for larger circuits [1]. Genetic algorithm based ATPG techniques were also proposed to find the multiple test sets in an efficient way [3]. Several approaches like ROBDD, parallel vector pair analysis were introduced to determine MSA faults [6][14][15].

The implicit ways of detecting the DSA using SSA vector set and to generate the extra test vectors for the undetermined faults have also been proposed in [7]. To find undetermined faults in an easy way is to go through all DSA faults and filter out the determined faults. But the complexity to detect n faults would be *O(mn)* which is again high.

The proposed algorithm considers path constraints violation as in one SSA fault checks for the other SSA fault that violates the path constraints and they are considered into a (PUDL) potentially undetected DSA fault list. The PUDL is passed over to find the faults in the form of $\{f_i,f_j\}$ and

$\{f_j,f_i\}$. Those faults are considered into UDL. This UDL is traversed to check if the DSA faults can be determined by the SSA test sets. The complexity of this algorithm is *O(m)*. This algorithm is similarly extended to the triple and quadruple faults [1].

A defect will be caused in any circuit during manufacturing. While applying test pattern to a circuit when the device is under testing, it gives output values of the circuit which if different from the fault free output values, indicate that the circuit has faults. These test patterns can be found by ATPG. The test pattern generation algorithms go by inducing the faults into the circuit and then activate the fault and make the induced value realize its effect at the primary output in the circuit, where it can be observed. The fault can be detected if the output responses of the faulty and fault free circuits differ [9][11][12].

ATPG is a procedure used to find a pattern succession which when applied to a digital circuit enables us to differentiate the fault free and faulty behaviour of circuit caused by defects while manufacturing. The parameters used to calculate the effectiveness of ATPG are: the number of generated vectors, fault models, number of faults detected, fault coverage [13]. ATPG application time increases with the circuit size and number of test patterns to be generated. Many ATPG methods have been developed over the years for combinational and sequential circuits such as D algorithm, PODEM algorithm, FAN algorithm [2] [7][8] [10].

III. PROPOSED WORK

Figure 1 shows the proposed algorithm to determine patterns for DSA, TSA and QSA faults using patterns for SSA faults. In this algorithm first the list of all DSA, TSA and QSA faults which cannot be detected using SSA is determined. An additional procedure is then used to obtain the patterns for the undetected faults. A multiple stuck-at fault occurring at a fan-out is very complicated. A test vector set that can detect all possible single stuck-at faults on the primary inputs and fan-outs of a circuit, can determine all single stuck-at faults in that circuit. The flow of the algorithm incorporates two fault filters. The step-by-step details of the proposed algorithm is as follows:

1. SSA fault list and their test set generation:

a. List of all the SSA faults and their test patterns are generated.

b. Consider only the primary inputs and the fan-out faults into the new SSA list which is in the SFL list.

2. Initial stage of fault filtering (DSA faults):

a. Consider an arbitrary list of Potentially undetected DSA fault list.

b. Consider an SSA fault as the fixed one from the SFL, now travel through the SFL to check for the path constraint violation group the respective faults and consider them into the Potentially Undetected DSA fault List (PUDL).

c. The aforementioned process is continued till the whole SFL is checked.

d. Pass over the PUDL and consider the faults which are in the form of $\{f_i,f_j\}$ and $\{f_j,f_i\}$ into the Undetected DSA fault list.

3. *Second fault filter of DSA faults*: This is done because there may be some DSA faults which can be determined by their respective test patterns or any other SSA test patterns. So in order to not generate unnecessary test patterns, the DSA faults that are detected are removed.

a. Go through the Undetected DSA List (UDL) to remove the detectable DSA faults.

b. Perform a simple ATPG approach to generate test sets for the remaining faults in the UDL.

A set of random patterns are generated for each undetermined DSA fault respectively. Where undetermined DSA faults are the ones which are not detected by SSA vector set. After the test patterns are generated and corresponding path constraints are found, we compare these path constraints with fault free path constraints and check the primary outputs of both the constraints. If there is a difference between them it means that the test pattern generated can detect the corresponding DSA fault. This process is repeated for all the undetected DSA faults. Following steps are used in the final test generation stage for DSA Faults.

For a particular DSA fault:

Authorized licensed use limited to: Rochester Institute of Technology. Downloaded on February 01,2023 at 16:54:03 UTC from IEEE Xplore. Restrictions apply.

1. If any of the faults in DSA pair is a fault at primary inputs, then we induce the values which are opposite of the stuck-at fault values of that particular SSA fault in the pair in their corresponding places.

2. If one of the faults in the DSA pair is a fault at fan-outs or else if both the faults in a DSA pair are faults at the fan-outs, then the opposite value of stuck-at fault value is induced while calculating the path constraints.

3. The remaining primary inputs which are not faults are generated randomly.

4. After test patterns are generated corresponding path constraints are found.

For Multiple stuck-at faults the same algorithm as of DSA faults is extended and implemented to generate test patterns.

1. *Finding triple stuck-at faults includes three cases*:

a. Three faults do not affect each other. In this case SSA test sets are sufficient to determine TSA faults.

b. Two of the three faults reciprocate the path constraints violation, and the other fault does not get affected by the remaining faults. In this case extra test patterns that are generated for DSA faults are sufficient to determine TSA faults.

c. Three faults reciprocate the obstruction of their propagation paths. The faults in this case are only needed to be examined.

2. A triple fault which includes three SSA faults r1, r2, r3, though {r1,r2} has a DSA test pattern generated but if the r3 is obstructing the pair and r3 is an undetermined fault or also obstructed by the {r1,r2} then an additional test set has to be generated for those faults. Now, consider all the aforementioned type of faults into a record and then generate the respective test sets.

3. Similar process is adopted to determine QSA faults.

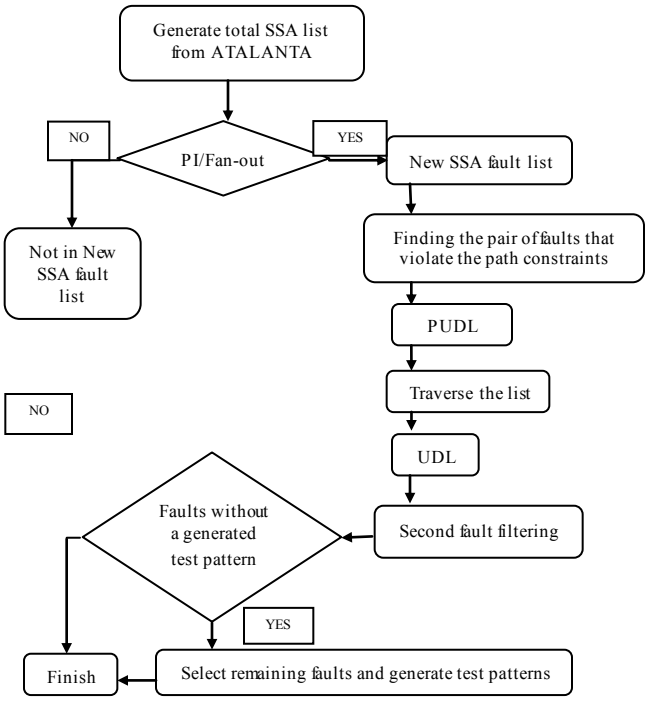

Fig.1. Proposed algorithm

The test patterns for the remaining faults are generated as same as for the DSA faults.

IV. EXPERIMENTAL RESULTS

The proposed ATPG algorithm was implemented in C language on a Linux Kernel 5.0 operating on Intel® Core™ i3-5005U CPU @ 2.00GHz × 4 and 7.7 GB memory. For SSA, the ATPG tool ATALANTA also generates compacted test patterns. The table 1 shows the number of SSA faults detected and their test patterns and the percentage of faults covered and the level of circuit for combinational circuits. The algorithm was applied on ISCAS'85 benchmark circuits.

Authorized licensed use limited to: Rochester Institute of Technology. Downloaded on February 01,2023 at 16:54:03 UTC from IEEE Xplore. Restrictions apply.

TABLE I. FAULTS AND TEST PATTERN SUMMARY OF ISCAS'85

Benchmark Circuit	*Count of DSA Faults*	*Count of detectable DSA Faults by test patterns of SSA Faults*	*Extra patterns generated to detect DSA Faults*
C17	64	12	52
C432	17205	1431	15774
C499	2764	203	2561
C880	7021	1016	6005
C1355	46053	525	45525
C1908	253	10	243

TABLE II. FAULTS AND TEST PATTERN SUMMARY OF DSA FAULTS

Bench mark circuit	*Number of SSA faults*	*Number of test patterns*	*Fault Cover age*	*Level of circuit*
C17	22	8	100	3
C432	524	75	99.23	17
C499	758	66	98.94	11
C880	942	115	100	24
C1355	1574	115	99.49	24
C1908	1879	179	99.52	40
C2670	2747	199	95.74	32
C3540	3428	244	96	47
C5315	5350	224	98.89	49
C6288	7744	58	96.51	124
C7522	7550	370	98.25	43

Table II gives the details of the whole count of DSA faults, number of detectable DSA faults by the test patterns of SSA faults and the extra test patterns required for each circuit to detect DSA faults. Now considering the total count of DSA faults for c17 benchmark circuit which is 64 faults with 12 DSA faults being determined by the generated SSA test patterns and 52 extra test patterns being generated to determine DSA faults. In the similar way C1355 benchmark circuit has 46053 DSA faults count out of which 525 DSA faults are being determined by the generated SSA test patterns and 45525 extra test patterns are being generated to determine the DSA faults.

TABLE III. FAULTS AND TEST PATTERN SUMMARY OF TSA AND QSA FAULTS

Benchmark Circuit	*Number of TSA Faults*	*Number of QSA Faults*
C17	960	2376
C432	4255370	49315207
C499	274018	1265625
C880	1109318	8214570
C1908	7590	10591

Table III gives an insight on the total count of Triple stuck-at faults (TSA) and Quadruple stuck-at faults (QSA) for each circuit.C17 benchmark circuit has 960 TSA faults and 2376 QSA faults. C432 benchmark circuit has 4255370 TSA faults and 49315207 QSA faults. Consider a DSA fault test pattern and find its path constraints. Now take an SSA fault test pattern and find its path constraints, compare both the path constraints, if there is any path constraint violation then group them as a TSA fault. Similarly other TSA faults are determined. For QSA faults consider a TSA fault test pattern and find its path constraints. Now take an SSA fault test pattern and find its path constraints, compare both the path constraints, if there is any path constraint violation then group them as a QSA fault. Similarly, all the other QSA faults are determined.

V. CONCLUSION AND FUTURE SCOPE

In this paper multiple stuck-at faults such as double, triple, quadruple faults are determined by the single stuck-at faults test patterns. This method or technique is implemented on various benchmark circuits. This methodology also includes the use of various tools like

ATALANTA. This algorithm is efficient and also generates compact test sets for multiple stuck-at faults. Future scope of the proposed method is this algorithm can be designed in a more efficient way to produce more compacted test sets in a very less runtime. This proposed idea can also be extended to various faults such as bridging faults, delay faults.

REFERENCES

[1] P. Wang, C. J. Moore, A. M. Gharehbaghi and M. Fujita, "An ATPG Method for Double Stuck-At Faults by Analyzing Propagation Paths of Single Faults," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 65, no. 3, March 2018 pp. 1063-1074.

[2] P. Wang, A. M. Gharehbaghi and M. Fujita, "Automatic Test Pattern Generation for Double Stuck-at Faults Based on Test Patterns of Single Faults," 20th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, USA, March 2019, pp. 284-290.

[3] J. P. Anita and P. T. Vanathi, "Genetic algorithm based test pattern generation for multiple stuck-at faults and test power reduction in VLSI circuits," 2014 International Conference on Electronics and Communication Systems (ICECS), Coimbatore, 2014, pp. 1-6.

[4] Yu Hu, Xiaowei Li, Wu-Tung Cheng,Yu Huang, and Huaxing Tang, 'Diagnose Failures Caused by Multiple Locations at a Time', IEEE Trans. on very large scale integration (VLSI) systems, vol. 22, no. 4, April. 2014.

[5] Stephan Eggrsglub, Kenneth Schmitz, Rene Krenz-Baath, and Rolf Drechsler."On Optimization-Based ATPG and Its Application for

Authorized licensed use limited to: Rochester Institute of Technology. Downloaded on February 01,2023 at 16:54:03 UTC from IEEE Xplore. Restrictions apply.

Highly Compacted Test Sets." IEEE Transactions on Computer Aided

[6] Mohan, N., & Anita, J. P. (2016). "A zero suppressed binary decision diagram-based test set relaxation for single and multiple stuck-at faults" .International Journal of Mathematical Modelling and Numerical Optimisation, 7(1), 83-96.

[7] Yi-Cheng Kung, Kuen-Jong Lee, and Sudhakar M. Reddy, 'Generating Single- and Double- Pattern Tests for Multiple CMOS Fault Models in One ATPG Run', IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems,2019.

[8] Masahiro Fujita, AlenMishchenko, 'Efficient SAT-based ATPG techniques for all multiple stuck-at faults', International Test Conference.

[9] Irith Pomeranz, Sudhakar M. Reddy,'Selection of a Fault Model for Fault DiagnosisBased on Unique Responses'. IEEE Trans. On Very Large Scale Integration (VLSI) Systems, vol. 18, no. 11, November 2010.

[10] RaimundUbar, Stephen AdeboyeOyeniran Tallinn, 'Multiple Fault Testing in Systems-on- Chip with High-Level Decision Diagrams',s10th International Design & Test Symposium,2015.

[11] Kumar, C. N., Madhumitha, A., Preetam, N. S., Gupta, P. V., & Anita, J. P. (2019, April). Fault Diagnosis Using Automatic Test Pattern Generation and Test Power Reduction Technique for VLSI Circuits. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 412-417).

[12] Mohan, Navya, And JP Anita. "Compact Test and Diagnosis Pattern Generation for Multiple Fault Pairs in Single Run." Journal of Engineering Science and Technology 15.6 (2020): 3820-3835.

[13] Hussain, Zakir, and J. P. Anita."Identification of faulty locations in digital circuits using SVM classifier." In 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), pp. 43-47. IEEE, 2020.

[14] S. Kajihara, R. Nishigaya, T. Sumioka and K. Kinoshita, "Efficient techniques for multiple fault test generation," Proceedings of IEEE 3rd Asian Test Symposium (ATS), Nara, Japan, 1994, pp. 52-56.

[15] S. Kajihara, T. Sumioka and K. Kinoshita, "Test generation for multiple faults based on parallel vector pair analysis," Proceedings of 1993 International Conference on Computer Aided Design (ICCAD), Santa Clara, CA, USA, 1993, pp. 436-439.

Authorized licensed use limited to: Rochester Institute of Technology. Downloaded on February 01,2023 at 16:54:03 UTC from IEEE Xplore. Restrictions apply.