P. Ajeesha Sulthana, Mr. A. Sivasankar M.E / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 4, Jul-Aug 2013, pp. 433-436

Test Vector Generation and Compression for Fault Detection Using NDR and RLE Algorithm

P. Ajeesha Sulthana[1], Mr. A. Sivasankar M.E[2]
[1](PG Scholar, Department of Electronics and Communication, Anna University: Regional Center, Madurai, Tamilnadu, India
[2](Assistant Professor, Department of Electronics and Communication, Anna University: Regional Center, Madurai, Tamilnadu, India

ABSTRACT

***N* detect tests are of mostly used because of their ability to improve the defect coverage. An *N*-detect test set detects each fault by at least *N* different vectors. The main problem which limits their use is their size. Researchers have proposed some methods to generate *N*-detect tests, but not much work has been done on compacting them. Also, there is no minimum size estimate of an *N*-detect test set. Methods for multiple detects test generation usually produce fully specified test patterns. This limits their usage in number of application such as low power test and test compression. Larger test data size demands not only higher memory requirements, but also increase the testing time. Test data compression overcomes this problem by reducing the test data volume without affecting the overall system performance. So this work provides a systematic methodology for identifying large number of unspecified bits in a multiple detect test sets, for detecting multiple faults while preserving the original fault coverage and the size of the test set is reduced using run length encoding method.**

Key words- Automatic Test Pattern Generation (ATPG), Multiple faults, N-detect test set, Run Length Encoding (RLE), Test set compaction

I. INTRODUCTION

In recent years, the development of integrated circuit technology has accelerated rapidly. VLSI techniques promise to make today's functional level devices and tomorrow's basic components. Accordingly, digital systems are built with more and more complexity. The fault testing and the diagnosis of digital circuits becomes an important and indispensable part of the manufacturing process. Since the cost of testing a VLSI chip is a significant fraction of the manufacturing cost. The time required to test a chip should be minimized, and there should be significant fault coverage. The objective of the automatic test pattern generator is to find an optimal set of test stimuli which detects all modeled faults, that is, a set of test input vector which when applied to the circuit can distinguish between the correct circuit and any circuit with a modeled fault.

II. TEST PATTERN GENERATION

In any circuit composed of logic gates, there is a possibility of occurrence of fault. A fault is defined to have occurred when a circuit variable assumes a value (1, 0 or X) which differs from that is violates the original circuit equation. ATPG is an electronic design automation method/technology used to find an input sequence that, when applied to a VLSI circuit, enables automatic test equipment to differentiate between the correct circuit behavior and the faulty circuit behavior caused by the defects.

The generated patterns are used to test the semiconductor devices after manufacture, and assist with determining the cause of failure. The quality of ATPG is measured by the amount of modeled defects, that are detected and the number of generated test patterns.

Current nanometer manufacturing processes suffer from larger defective parts ratio, partly due to numerous emerging defects. The traditional fault models, such as the stuck-at and transition delay fault models are still widely used, they have been shown to be inadequate to handle these new defects. One possible solution to this problem is to develop complex fault models to imitate defect behavior at either the logic or layout level of abstraction. The combination of the large number of possible defect types together with the huge number of fault sites in a modern circuit implies that modeling these defects will give prohibitively large input for a systematic test generation methodology [1],

In [2] Rapid increase in population increased the usage of digital components dramatically. For profitable income, the cost of the finished product and time taken for marketing the product needs to be reduced. In this paper [3] we illustrate that the traditional N detect ATPG is un optimized in terms of the size of the generated pattern set. The optimization problem is modified as a minimum covering problem. The Integer Linear Programming (ILP) is applied to obtain an N detection ATPG pattern set with the minimum number of patterns.

In [4] As the complexity of integrated circuits has increased, so it is need for improving testing efficiency. The types of defects are also

becoming very complex, which in turn makes simple approaches for testing inadequate. By the use of n-detect testing can improve detect coverage; however, this approach can increase the test set size. This proof-of-concept paper investigate the use of logic implication checkers, which is inserted in hardware, for compacting n-detect test sets.

In [5] this proposed procedure accepts a one-detection test set. This paper extracts test cubes for target faults from the one-detection test set. Then it merges the cubes in different ways to obtain an *n* -detection test set. In this paper, the authors conducted extensive survey of methods developed earlier to detect faults and minimize test set in digital circuits. The important power reduction may be obtained when appropriately fixing the unspecified bits. The work in this paper considers the problem of relaxing an n-detect test set. In [6] this can greatly increase the size of test set. Here we investigate the use of logic implication of checkers in hardware as an aid in compacting n-detect test sets.

In [7] a test set for stuck at faults of a combinational circuit or a full scan sequential circuit, values may be changed to the opposite logic values without losing fault coverage. Such input values can regarded as don't care(X). This method use fault simulation and procedures similar to implication and justification of ATPG algorithms

In [8] Rapid increase in population increased the usage of digital components dramatically and their production. To have the profit the cost of the finished product and time taken for marketing the product should be reduced. Extensive survey shows that new methods are developed to detect faults and minimize test set in digital circuits. In [9] this paper, we illustrate that the traditional N detect ATPG is unoptimized in terms of the size of the generated pattern set. Integer Linear Programming (ILP) is applied to obtain an N detection ATPG pattern set with the minimum number of patterns. In [10] defect oriented testing in digital circuits is a hard process. The proposed method also applies to multiple detect test sets (where, instead of, a variable number of tests exists per fault), but, without any loss of generality, we present it here only for -detect test sets. Whenever, necessary we elaborate on the trivial modifications that must be made for multiple detect test sets.

The method starts with an initial (given) test set which can be fully or partially specified. The total number of specified bits in the resulting test set is minimized, while maintaining its original -detect fault coverage. Moreover, the test set size is guaranteed not to increase; actually, it is often decreased. The motivation behind this problem is that a test bit needs to be initially fixed only if this helps the n-detect fault coverage otherwise the bit can be left unspecified. The generated relaxed test set can then be used in a variety of applications that fix the unspecified bits.

III. TEST DATA COMPRESSION

Data compression involves encoding information using fewer bits than the original representation. Compression can be either lossy or lossless. Lossless compression reduces bits by identifying and eliminating statistical redundancy. But no information is lost in lossless compression. The run length encoding is a lossless compression.

The sequence of length l of a repeated '0' is replaced by a shorter sequence, usually containing one or more symbols of '0', and sometimes an escape symbol. RLE algorithms differ from each other mainly in three points: the threshold t, the marking of the start of a run and the coding of the length information. If l is smaller than t, the run stays unchanged, if l is greater or equal t, the run is replaced. The start of a run can be indicated by a threshold run or an escape symbol 'c'. If a threshold run is used, the start is characterized by a small sequence of '0', which has a length greater or equal than t. If an escape symbol 'c' indicates the start of a run, 's' is normally put behind 'c' in order to characterize the run symbol. 'c' must not be an element of the alphabet or occurrences of 'c' have to be coded in a way, that they are not mixed up with the start of a run.

IV. PROPOSED WORK

Proposed work targets an test set optimization problem. This work determines multiple test to detect multiple faults that give the maximum benefit in terms of specified bit savings in the entire test set. Thus it select the best n test to detect faults and drops the faults from the remaining tests in order to reduce the total number of specified bits in this tests. Focus on multiple faults.

In n detect relaxation unit the generated test vectors of multiple faults are given as input. The N-detect or multiple detect test sets employs conventional fault models (stuck-at), typically without the use of layout information to generate patterns by targeting faults more than one time to increase the probability of catching non modeled defects.

Non-drop fault simulation is performed to determine the natural N-detect profiles of the tests. Faults with N-detect coverage lower than a threshold (N) are targeted for incremental N-detect test generation. The threshold is set to 10 in this work since it is expected that there would be diminishing return beyond this point. Further, it was estimated that test application (data volume and test time) and test generation constraints would prevent the generation and inclusion of the patterns beyond this threshold.

A simple definition of N-detect where only pattern difference counted towards number of detections is used in the N-detect test generation process.

In the algorithm used, first the input circuits will be given. Then the corresponding fault lists are generated. Test sets are formed for the generated fault lists. The unspecified bits will be calculated from the test sets. The maximum gain is calculated for the corresponding fault. This process is iterated for all faults. Once all the test sets are verified, then the fault is dropped from other tests.

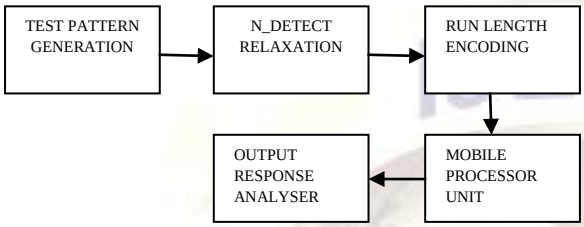

Fig 1: Proposed Work Block Diagram

The problem of detecting faults in multilevel circuits is considerably more complicated than in the case of two-level circuits. Except in the case where each gate has only a fan out of 1, it is no longer true that testing only the inputs will always detect all the faults within the circuit.

A fault in one path may not always be detectable if the other path is faultless. It shall be concerned with procedures for the detection of single faults.

This limitation does not, exclude the detection of most double and other multiple faults, but it emphasizes that only single faults will be detected in all cases, while some multiple faults may not be detected. The cross checking unit contains the details about various faults and test vector generation, which includes fault model, power requirement and so on. It will compare the behavior of the modeled fault with that of the original fault for verification.

Then that test vectors are given to the run length encoding process, where the test data going to be compressed. Test data compression involves adding some additional on-chip hardware before and after the scan chains. The additional hardware used will de-compresses the test stimulus coming from the tester and permits storing of test data in a compressed form on the tester. With test data compression, the tester still applies a precise deterministic test set to the circuit under test (CUT).

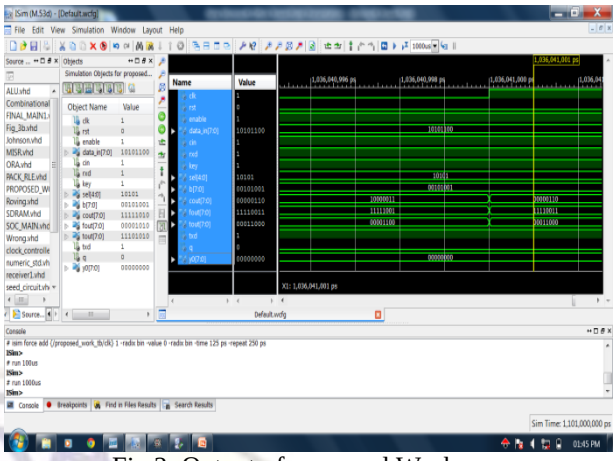

Fig 2: Output of proposed Work

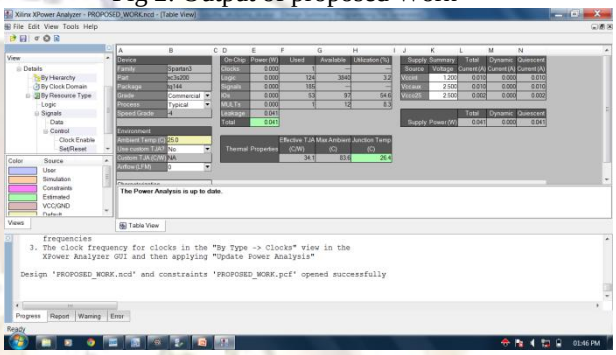

Fig 3: Power Analysis of Proposed Work

V. CONCLUSION

Using N_detect Algorithm, the test set for multiple faults were generated .The test set compaction of generated test vectors also carried out. The test data were compressed using run length encoding The algorithm was applied to mobile processor benchmark circuit and the results were analyzed. The power reduction also achieved. The experimental results reported demonstrate the effectiveness of the proposed method in achieving high specified bit reduction rates in –detect test sets, while maintaining the multiple detect fault coverage. In future, the algorithm is implemented in FPGA kit and the practical issues will be analyzed further.

REFERENCES

[1] Stelios N. Neophytou, and Maria K. Michael, Test Pattern Generation of Relaxed *n*-Detect Test Sets, *IEEE Transactions On Very Large Scale Integration Systems, Vol. 20, No. 3, March 2012*

[2] Karthikeyan A, Rajeswaran. N, Design And Implementation Of Multiple Fault Diagnosis On VLSI Circuits Using Artificial Neural Networks, *International Journal of Advances in Engineering & Technology, May 2012. vol. 3, Issue 2, pp. 685-695*

[3] S.M.Thamarai, Dr.K.Kuppusamy, Dr.T.Meyyappan, Fault Detection and Test Minimization Methods for Combinational Circuits - A Survey, *International Journal of Computer Trends and Technology- volume2Issue2- 2011*

[4] N. Alves, J. Dworak, I. Bahar, and K. Nepal, Compacting test vector sets via strategic use of implications, in *Proc. ICCAD*, 2009, pp.83–88.

[5] Pomeranz I. and S. M. Reddy, Forming n-detection test sets without test generation, *ACM Trans. Design Autom. Electron. Syst.*, vol. 12, no. 2, Apr. 1, 2007, Article 18

[6] Kantipudi, K. ,On the size and generation of minimal n-detection tests, in *Proc VLSID*, 2006, pp. 425–430.

[7] Vaidya B. and M. Tahoori, Delay test generation with all reachable output propagation and multiple excitation proc, in *Proc. DFTS*, 2005, pp. 380–388

[8] K. Miyase and S. Kajihara, XID: Don't care identification of test patterns for combinational circuits, *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*,vol. 23, no. 2, pp. 321–326, 2004.

[9] McCluskey E. J. and C.-W. Tseng, Stuck-fault tests vs. actual defects, in *Proc. ITC*, 2000, pp. 336–343.

[10] Butler .K and M. Mercer, The influences of fault type and topology on fault model performance and the implications to test and testable design, in *Proc. DAC*, 1990, pp. 673–678.