International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:268
Vol.2, No.4, April 2013 DOI:10.15693/ijaist/2013.v2i4.170-176

Test Pattern Generation for Relaxed n-detect test sets using Genetic Algorithm

Nalini.S
PG scholar
Karpagam university
Coimbatore-21

Sivakumar.S
Assistant professor
Karpagamuniversity
Coimbatore-21

Abstract---While defect oriented testing in digital circuits is a hard process, detecting a modeled fault more than one time has been shown to result in high defect coverage. Previous work shows that such test sets, known as multiple detect or n–detect test sets, are of increased quality for a number of common defects in deep sub-micrometer technologies. Method for multiple detect test generation usually produce fully specified test patterns. This limits their usage in a number of important applications such as low power test and test compression. This work proposes a systematic methodology for identifying a large number of bits that can be unspecified in a multiple detect (n -detect) test set, while preserving the original fault coverage. The experimental results demonstrate that the number of specified bits in, even compact, n-detect test sets can be significantly reduced without any impact on the -detect property. Additionally, in many cases, the size of the test set is reduced. A prototype system named GATTO is used to access the effectiveness of the approach in terms of result quality and CPU time requirement. The flexibility of GATTO enables users to easily tradeoff fault coverage and CPU time to suit their needs.

Keywords*---LFSR, Automatic test pattern generation(ATPG),Digital Circuit testing(DCT).*

I. INTRODUCTION

A. Motivation of the Project

Previously proposed methods for deriving single-detect relaxed test sets can be categorized into static or dynamic. Static methods consider an initial test set whereas dynamic methods incorporate the problem in the ATPG process. Extending these methods to n -detect test sets is not straightforward. Actually, the static methods benefit from identifying essential tests (a test is essential if it is the only one detecting a fault) which do not exist in n-detect test sets. Moreover, the common underlying idea used in both types of methods is the identification of coincidental multiple times fault detections, i.e., a fault is detected by several different tests even though it was only targeted once in the test generation process. The latter occurs very often, especially in the traditional stuck-at fault model, because the majority of the faults are easy-to-detect (also referred to as randomly detect faults). The symbolic approach is the best known solution for circuits with a limited number of state elements, while larger Sequential circuits still constitute a problem. The topological approach does not have similar limitations, but due to excessive backtracking, it often requires unacceptable CPU times when the circuit is too large or complex

B. Objective of the Project

The work in this paper considers the problem of relaxing an n-detect test set. The proposed method also applies to multiple n-detect test sets (where, instead of n, a variable number of tests exists per fault), but, without any loss of generality, we present it here only for n-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. Furthermore, 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 it can be left unspecified. The generated relaxed test set can then be used in a variety of applications that fix the unspecified bits appropriately. The applied fully specified test set is expected to have similar defect and non-targeted fault coverage to that of the original test set. The goal of this paper is to present a new approach combining the advantages of the adoption of a suitable technique based on Genetic Algorithm and of its extension to parallel and distributed systems.

II. TEST PATTERN GENERATION OF RELAXED n – DETECT TEST SETS

The work in this paper considers the problem of relaxing an n-detect test set. The proposed method also applies to multiple n-detect test sets (where, instead of n, a variable number of tests exists per fault), but, without any loss of generality, we present it here only for n-detect test sets. Whenever, necessary we elaborate on the trivial modifications that must be made for multiple detect test sets.

International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:268
Vol.2, No.4, April 2013 DOI:10.15693/ijaist/2013.v2i4.170-176

A. Multiple Detect Test

Current nanometer manufacturing processes suffer from larger defective parts ratio, partly due to numerous emerging defect types. While 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.

The influences of fault type and topology on fault model performance and the implications to test and testable design. Moreover, detailed layout information is typically not available until the fabrication phase, giving limited information to test engineers. Instead, previous work proposed the use of test sets targeting each modeled fault multiple times in order to increase the probability of detecting additional fault types as well as the defect coverage Impact of multiple- detect test patterns on product quality, Stuck-fault tests vs. actual defects.

Test sets detecting each fault with n different tests are known as n-detect test sets. A variety of multiple detect or n – detect test set generation methods and their impact in the quality of the testing process have been proposed Impact of multiple- detect test patterns on product quality. Some of these methodologies propose approaches that use metrics obtained by the probabilistic distribution that random defects are expected to follow.

B. Relaxed n –Detect Test Set

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. Furthermore, 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 it can be left unspecified. The generated relaxed test set can then be used in a variety of applications that fix the unspecified bits appropriately. The applied fully specified test set is expected to have similar defect and non-targeted fault coverage to that of the original test set. This is justified by the simple observation that in existing test generation techniques considering the traditional n-detect fault definition, any improvement in the coverage of non-targeted faults and defects, beyond the -detect per targeted fault improvement, is caused by the random bit fixing. The latter is supported by experimentally obtained data. The following circuits are simulated and generate the test pattern.

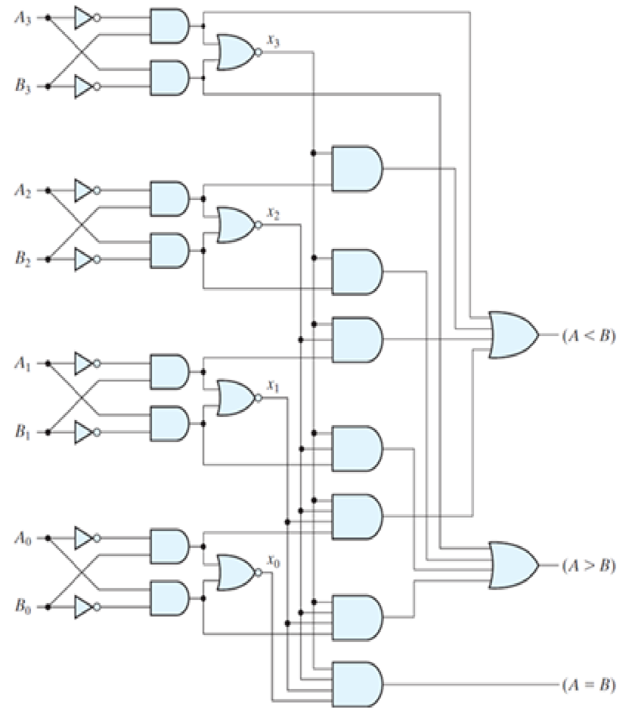

Fig. 1 Magnitude Comparator Circuit for Test pattern generation

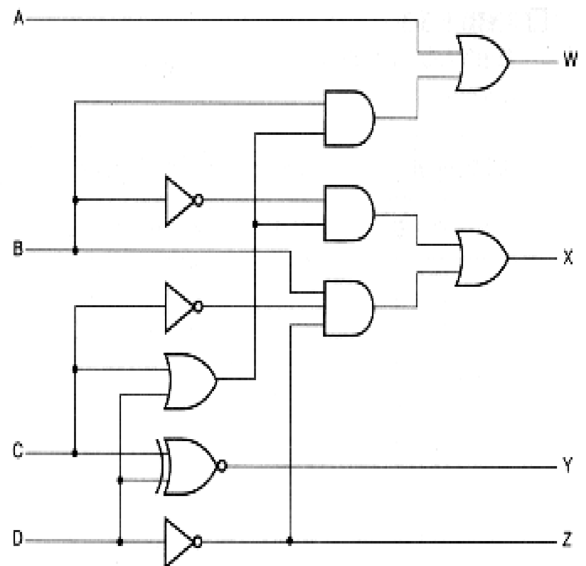

Fig. 2 BCD to Excess -3 convertor Circuit for Test pattern generation

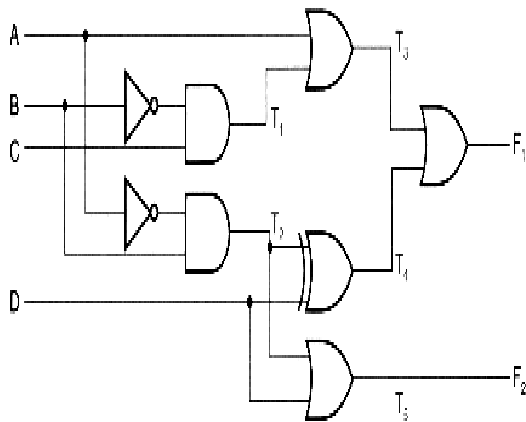

Fig. 3 Priority Encoder Circuit for Test pattern generation

C. Problem Formulation and Notation

Consider a given n-detect test set T= {t1, t2,….tm} for a combinational or a fully-scanned sequential circuit-under-test C. Each of the m test patterns consists of strings of three-valued bits E {0, 1, x}. Consider also a fault model, based on which the list of faults detected by T, denoted by F , is derived. For the considered fault list, the test set T has n-*detect fault coverage*, denoted by N(T) and *specified bits ratio*, denoted by K(T), as defined below. *Definition :* The n-detect fault coverage of a test set T, denoted by N(T) is the percentage of the faults considered, under a given fault model M with fault list F = {f1, f2,….,fk} that are detected by T by with at least n different tests.

Note, that for some faults only 0<p<n different tests exist (i.e., can be generated by any test generation process). Let Fn denote the fault list of all faults in F that are detected at least n times in T. Similarly, let Fp denote the faults in F that are detected less than n times in T. Clearly,Fn U Fp =F. In the case where no tests exist for a fault fi , then fault fi is considered to be redundant. In this work, N(T) is calculated considering all faults in F. *Definition :* For a test set T , the ratio of the bits having a specified value {0,1} over the total number of test set bits is defined as the specified bits ratio, denoted by K(T) .This ratio gives a test set property that indicates how flexible a test set is. Clearly,0<K(T)<1, for any test set. The closer K(T) is to 0, the more flexible T is. For fully specified test sets, K(T) = 1.

The above constraints give the specifications of the test set relaxation problem considered. Constraint (i) preserves the n-detect fault coverage in the same way as the fault coverage is preserved in single detect test sets. If a fault fi is detected p<n times in test set T (i.e.Dfi = p< n), then fault fi is also detected p times in the relaxed test set T'. All other faults, i.e.,those with Dfi > n , are detected at least n times in the relaxed test set. Increase of the fault coverage of T' may occur due to coincidental detections for any fault fi .Constraint (ii) guarantees that the relaxed test set has no more tests than T , preventing the relaxation process from increasing the size of the test set in an attempt to give a higher number of unspecified bit in the test set. Constraint (iii) comes directly from the definition of the relaxation problem since the overall goal is to decrease the portion of specified bits in the test set.

D. Theoretical Framework

In the proposed method, every test in T is systematically replaced by a new test with more unspecified bits. The algorithm concentrates on one fault at a time to determine n different tests {tj,j= 1,2,…n} that detect the fault such that the number of bits that can be relaxed in the entire test set is maximized. Put differently, the algorithm determines n tests to explicitly target the fault and relaxes the bits required to detect the fault in the remaining tests. We first give the theoretical framework and then we present in detail all the steps of the proposed methodology. Consider a fault fi detected by T. Let Ti<T denote the set of tests in that detect fault fi. The algorithm finds the n tests in Ti , given in that should detect fault . Consider a test. Let the number of specified bits in tk that can be unspecified if tk no longer detects fi be denoted by Cik. In other words, is the contribution, in specified bits, of fault fi in test tk. Then, the total number of specified bits in that can become unspecified if fault is only detected by test (and not by any other in{Ti – Tj})

It can be argued that, since we want to find the number of test set bits that can become unspecified after keeping only detections, should take into consideration all combinations of detections. In other words, it is a question whether keeping the detections that give the larger specified bits relaxation is as effective as keeping the combination of detections that give the larger relaxation. Next, we prove that the two decision criteria are identical. First we slightly modify equations in order to evaluate the gain and maximum gain in specified bits considering all combinations of detections for the same fault.

III PROPOSED ALGORITHM

A. Genetic Algorithm

GA's have been already considered as a possible ATPG technique in [9], [10], and [11]. In the first paper, the CRIS ATPG has been proposed, which generates test vectors on the basis of an *objective function* measuring the extent to which the activity is balanced within the circuit. The method considers both combinational and sequential circuits, and logic simulation is used to compute the objective function. By comparison with *[9],* the approach presented here focuses only on sequential circuits and adopts techniques especially suited for them, resulting in a comparable efficiency but shorter test sequences. In [10], the same authors propose the integration of

CRIS with a topological algorithm in order to combine the advantages of the two approaches: a GA is used for fast test vector generation, and a topological one is activated to escape from local minima and to identify investable faults.

The method described in [11] is based on a *fitness function* which exploits the results predicted by a fault simulator; one vector at a time is generated, following a three phase approach, as in CONTEST [3]. When the method is no longer able to increase the fault coverage by generating test vectors, it switches to generation of test sequences. Our approach is also organized in three phases, but the meaning of the three phases is completely different. Only one phase is implemented as a GA, and its goal is the detection of a specific fault, instead of the increase of the total fault coverage. Moreover, our method always deals with test sequences instead of test vectors. Our approach primarily aims at providing an effective solution for the circuits for which the symbolic techniques are not able to provide any result, but is also able to deal with all the standard benchmark circuits with acceptable CPU time requirements.

Its main advantage is the provision of good results with acceptable CPU times, and it allows the user to easily tradeoff fault coverage and CPU time. We devised a distributed version of the approach to further enhance its performance by exploiting the computational power of the workstation networks which are now available in most design centers. We adopted a portable message-passing library which makes full use of their computational and memory resources without being bound to any specific hardware platform. Our paper demonstrates that the combination of an efficient simulation-based ATPG technique with the power of such networks is both feasible and profitable. A description will now be given of our prototype algorithm named GATTO (Genetic Algorithm for Test pattern generation). In particular, Section I1 briefly outlines GA' s; Sections Ill and IV describe the monoprocessor algorithm and its distributed version GATTO", respectively. Section V draws some conclusions.

GA's are evolutionary algorithms that mimic the way nature improves the characteristics of living beings. Each solution (individual) is represented as a string (chromosome) of elements (genes) and is assigned a fitness value based on the value given by an evaluation function. The evaluation function measures how close the individual is to the optimum solution. A set of individuals constitutes a population that evolves from one generation to the next through the creation of new individuals and the deletion of some old ones. The process starts with an initial population created in some way, e.g., randomly. Evolution can take two forms.

- **Crossover:** The chromosomes of two individuals are combined to obtain a new individual for insertion in the population to replace another individual, e.g., the one with the lowest fitness (elitism). The crossover is generally backed by techniques ensuring that the selection probability of each individual is proportional to its fitness. New individuals are thus likely to have a higher fitness than those they replace. The process is oriented toward those regions of the search space where optimal solutions are supposed to exist.
- **Mutation:** A gene of a selected individual is randomly changed. This provides additional chances of entering unexplored regions. Evolution is stopped when either the goal is reached or a maximum CPU time has been spent.

B. Applications of LFSR

In general, xors are only ever 2-input and never connect in series. Therefore the minimum clock period for these circuits is T > T2, input-xor clock overhead, Very little latency, and independent of n. This can be used as a fast counter, if the particular sequence of count values is not important. For example micro-code, micro-pc.It Can be used as a random number generator. Sequence is a pseudorandom sequence numbers appear in a random sequence. Random numbers useful in computer graphics, cryptography, automatic testing, • Used for error detection and correction, CRC (cyclic redundancy codes) ethernet uses them.

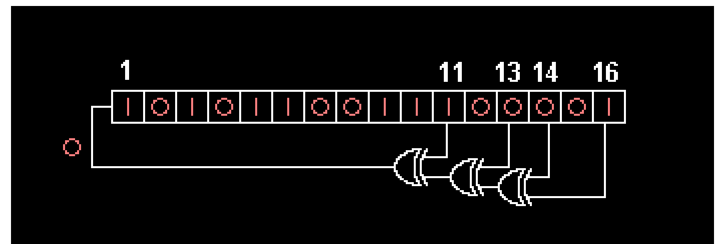

Fig. 4. Linear feedback shift register

VI. Results and Discussions

In this project we have developed software to continuously generate test patterns to detect the faults in the circuit. We have developed a Test Pattern Generation of S27, S208 and S298 Circuits. And we have developed software using VHDL coding and implemented by using CADENCE software. Also the power calculation is done by using the VHDL coding and conducted by using CADENCE Tool. Test pattern generation and n-detect test set output waveforms are given below.

A. Output Waveform for Test Pattern Generation of Relaxed n-detect test set using Genetic Algorithm

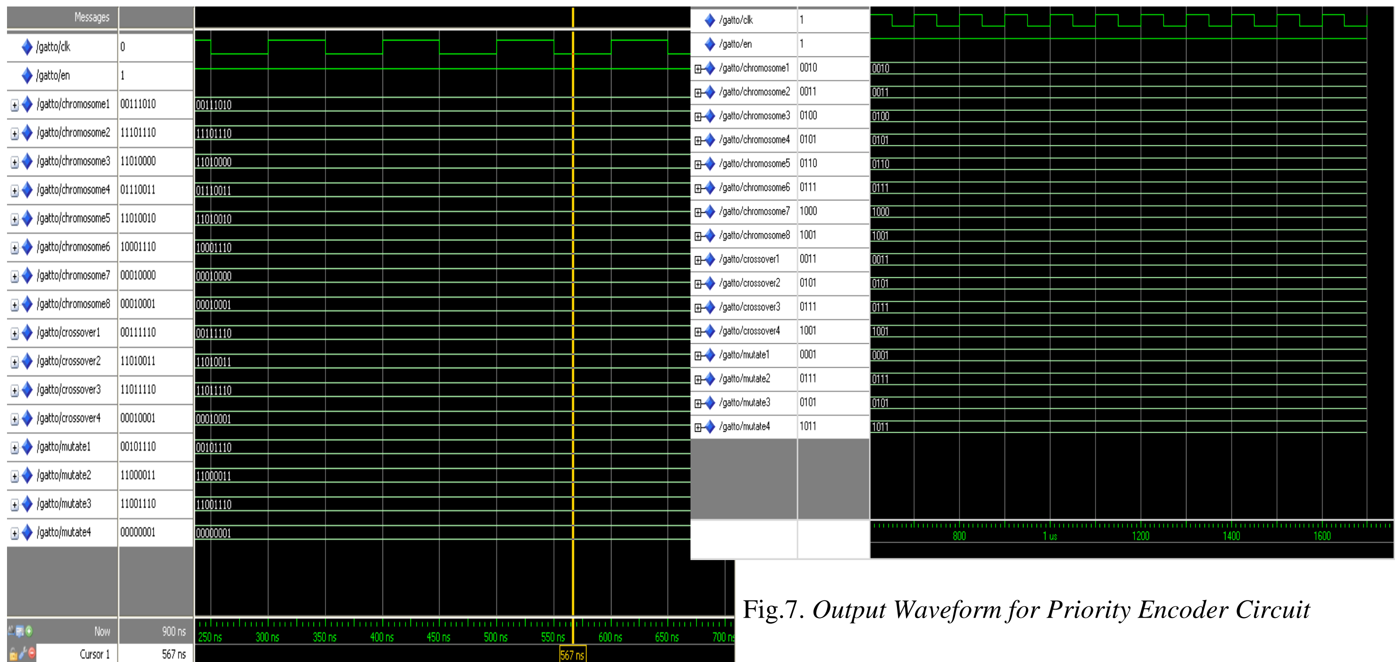

Fig. 5. *Output Waveform for Magnitude Comparator Circuit*

Fig.7. *Output Waveform for Priority Encoder Circuit*

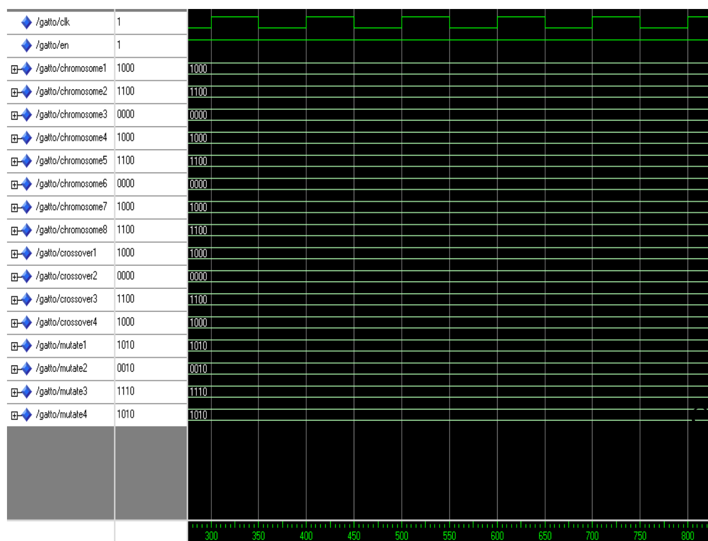

Fig. 6. *Output Waveform for BCD to Excess3 Convertor Circuit*

B. Power Report

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		68
Vccint 1.80V:	34	61
Vcco33 3.30V:	2	7
Clocks:	18	33
Inputs:	1	2
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	15	27
Quiescent Vcco33 3.30V:	2	7

Thermal summary:	
Estimated junction temperature:	26C
Ambient temp:	25C
Case temp:	26C
Theta J-A:	17C/W

valid Program Mode

Decoupling Network Summary:	Cap Range (uF)	#

Fig.8. Power Report for Magnitude Comparator Circuit

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		70
Vccint 1.80V:	35	63
Vcco33 3.30V:	2	7
Clocks:	19	34
Inputs:	1	2
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	15	27
Quiescent Vcco33 3.30V:	2	7

Thermal summary:	
Estimated junction temperature:	26C
Ambient temp:	25C
Case temp:	26C
Theta J-A:	17C/W

Invalid Program Mode

Decoupling Network Summary:	Cap Range (uF)	#
Capacitor Recommendations:		

Fig. 9. Power Report for BCD to Excess3 Circuit

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		64
Vccint 1.80V:	32	57
Vcco33 3.30V:	2	7
Clocks:	16	29
Inputs:	1	2
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	15	27
Quiescent Vcco33 3.30V:	2	7

Thermal summary:	
Estimated junction temperature:	26C
Ambient temp:	25C
Case temp:	26C
Theta J-A:	17C/W

Fig. 10. Power Report for Priority Encoder Circuit

TABLE I
Experimental Result for Our Approach

S.NO	CIRCUITS	INPUTS	EXISTING METHOD	PROPOSED METHOD
1	Magnitude Comparator	10	8	4
2	BCD to Excess3 Convertor	4	8	4
3	Priority Encoder	4	8	4

1) Conclusion

In this work, we have investigated the impact of test set relaxation in n-detect test sets using Genetic Algorithm. We presented a systematic methodology for decreasing the number of specified bits in a given n-detect test set or a multiple detect test set by using Genetic Algorithm. The experimental results reported demonstrate the effectiveness of the proposed method in achieving high specified bit reduction rates in n-detect test sets, while maintaining the n-detect fault coverage. Provided discussion and experimentation data also explain how the defect coverage and non-targeted fault coverage of the relaxed test sets using Genetic Algorithm are less. will be similar to that of the initial test sets, when the relaxed test sets are fully-specified before test application. The flexibility of GATTO enables users to easily tradeoff fault coverage and CPU time to suit their needs.

Reference

[1] J. Dworak, J. D. Wicker, S. Lee, M. R. Grimaila, M. R. Mercer, K.M. Butler, B. Stewart, and L.-C. Wang, "Defect-oriented testing and defective-part-level prediction," *IEEE Design Test Comput., vol. 18,*no. 1, pp. 31–41, Jan. 2001.

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

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

[4] Y. Huang, "On n-detect pattern set optimization," in *Proc. ISQED*, 2006, pp. 445–450.

[5] K. Kantipudi and V. Agrawal, "A reduced complexity algorithm for minimizing n-detect tests," in *Proc. VLSID*, 2007, pp. 492–497.

[6] E. Kalligeros, X. Kavousianos, and D. Nikolos, "Multiphase BIST: A new reseeding technique for high test-data compression," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 10, pp. 1429–1446, 2004.

[7] S. Neophytou and M. Michael, "Test set generation with a large number of unspecified bits using static and dynamic techniques," *IEEE Trans. Comput., vol. 59, no. 3, pp. 301–316, Mar. 2010.*

[8] Ya-Ting Shyu, Jai-Ming Lin, Chun-Po Huang, Cheng-Wu Lin, Ying-Zu Lin, and Soon-Jyh Chang *Member IEEE* "Effective and Efficient Approach for Power Reduction by Using Multi-Bit Flip-Flops", IEEE transactions on very large scale integration (vlsi) systems, This article has been accepted for inclusion in a future issue of this journal.

[9] D. G. Saab, Y. G. Saah, and J. Abraham, "CRIS: A test cultivation program for sequential VLSI circuits," in *Proc. Int. Conf Computer- Aidid Design,* i992, pp. 216-219.

*[10]*D. G. Saab, Y. G. Saah, and J. Abraham, "Iterative [simulation-based + deterministic techniques] = complete ATPG," in *Proc. Int. Con$*

[11] E. **M.** Rudnick, J. H. Patel, G. S. Greenstein, and T. M. Niermann,"Sequential circuit test generation in a genetic algorithm framework,"in *Proc. Design Automation Con$,* 1994, pp. 698-704.

Authors Profile

S.SIVAKUMAR is an Assistant Professor, Dept. of Electronics and Communication Engineering, Karpagam University, Coimbatore, India. He obtained his Bachelor degree in Electrical and Electronics Engineering from

Coimbatore Institute of Technology and Master degree in VLSI Design from Anna University Coimbatore, India. His research interests include Low power VLSI and IC Fabrications.

S.NALINI received the **B.E.** degree in electrical and electronics engineering from the Bannariamman Instotute of technology, Sathiyamangalam, Anna University, Chennai, India, in 2007.Currently doing **M.E.** in VLSI design in Karpagam University Coimbatore, India. Her research interest includes digital design and low power vlsi.