

Analysis and Detection of various Faults in Combinational Circuits using D-Algorithm and BIST

¹Umesh

II year, M.Tech (Digital Electronics)
Department of E&CE

S. D. M. College of Engineering and Technology,
Dharwad, Karnataka

Email ID: umesh.gavaral@gmail.com

²Kotresh E. Marali

Assistant Professor,
Department of E&CE

S. D. M. College of Engineering and Technology,
Dharwad, Karnataka

Email ID: kmarali18@gmail.com

Abstract— After a digital circuit has been designed; it is fabricated in the form of silicon chips. The fabrication process is not perfect and due to various reasons, the manufactured circuit in silicon may develop defects which may prevent its correct functioning. A manufacturing test performs the crucial task of identifying those silicon chips that do not function as expected. It involves exercising the functionality of the Circuit Under Test (CUT) by applying appropriate test signals to its inputs and observing the responses. If the responses of the CUT match the expected responses, then the CUT is considered good else it is labeled as bad. Thus, the goal of testing is to correctly identify a good chip as good and a bad chip as bad. The testing process may not be perfect and it may label certain good chips as bad and is known as yield loss and it may label certain bad chips as good and is known as test escapes. The yield loss results in economic loss due to throwing away a proportion of good chips. Test escapes, on the other hand, result in defective parts shipped to customers and, depending on the application, have moderate to serious consequences in terms of system failures, economic damages, etc. The testing process thus needs to ensure that both of these proportions are kept to a minimum. Hence the validation and verification of the integrated circuits (IC's) is very much essential before it is manufactured. In this regard an attempt is made in realizing the occurrence of various faults in digital circuits such as single stuck-at faults and detecting the same. The most widely known gate-level test generation algorithms are the D-algorithm and PODEM (Path Oriented Decision Making) and BIST. However, the fault analysis in these circuits is made by injecting the faults, which in turn validated by some techniques such as D-algorithm and BIST.

Keywords— *Circuit Under Test (CUT), Yield Loss, Test Escapes, Integrated Circuits (IC's), single stuck-at faults, D-algorithm, BIST. Introduction (Heading 1)*

1. INTRODUCTION

When digital circuits are fabricated in the form of silicon chips, due to various fabrication process aberrations, some of the chips develop defects which may prevent their correct functioning. It is the goal of manufacturing testing to determine whether a chip possesses any such fault-causing defects, in a given finite time allotted for testing.

The testing of integrated circuits is an important part before it is coming to the market; hence the fault simulation is required for testing these IC's. Test generation is the most important step in manufacturing testing in which, given a set of faults defined using a fault model, appropriate test signals, called test vectors, are generated, which when applied to the CUT are able to detect the presence of those faults. The

program which generates these test vectors is called an Automatic Test Pattern Generator (ATPG). As a means to increase the testability of the circuits and also to reduce the Automatic Test Pattern Generation (ATPG) complexity, Design-For-Test (DFT) methods are employed. Two main parameters that determine the testability of a circuit are the controllability and observability of its signals. Controllability of a signal refers to its ability or ease to be set to a particular logic value from the primary inputs of the circuit. Observability of a signal refers to its ability or ease to be observed at one of the primary outputs of the circuit. Design-for-test (DFT) method refers to the design method of improving the controllability and observability of the signals of the given digital circuit so that the overall testability of the circuit is enhanced and tests with high fault coverage can be derived in reduced time complexity. Several DFT schemes are employed in practice and one such method is Fault modeling. In fault simulation, the test vectors are simulated on the CUT in the presence of one fault at a time and the response of the CUT to the test vectors is compared with the expected correct responses. If the simulated responses differ from the expected correct responses, then the fault being simulated is considered to be detected, this process is known as verification. Along with evaluating the effectiveness of the test vectors, fault simulation also forms an integral part of the ATPG program. By comparing the time complexity for solving the test generation problem with that of fault simulation, we can deduce that fault simulation based test generation methods can provide lower time complexity.

A. Fault Modeling

Faults at the physical level in chips cannot be tested and detected directly, as there could be numerous types that can occur and many of them are often complex in nature to analyze. Hence faults need to be modeled at a higher abstraction level in order that they can be analyzed and test signals generated to detect them. These models are generally referred to as fault models. Faults can be modeled at various abstraction levels starting with the lowest level like the transistor and gate level; and moving to higher levels like Register-Transfer-Level (RTL) and behavioral level. Based on these abstraction levels, the fault models can be roughly classified as Lower-level fault models and Higher-level fault models.

B. Lower-level Fault Models

The lower-level fault models include those defined at the transistor and the gate levels. At this abstraction level, the digital design is described as an interconnection of transistors and gates, and faults can be modeled as imperfections in their respective components. Some of the commonly used and popular fault models at the transistor and gate-level are stuck-at fault model, transition delay fault model and IDDQ fault model

- *Stuck-at fault model*

One of the most widely used fault models for gate-level digital is the stuck-at fault model. The faults are modeled on signal lines or interconnect between the gates. Using the stuck-at fault model, two types of faults can be modeled for any signal line in the gate-level digital circuit. The logic value of a considered faulty signal line could be permanently stuck-at logic '0' or stuck-at logic '1'. Figure 1.1 shows the behavior of a stuck-at logic '0' on the output of the AND gate. Stuck-at faults model some of the physical defects that could arise in silicon manufacturing like transistors permanently in ON or OFF state, shorting of signal lines to power supply lines (VDD: logic '1' or GND: logic '0'), etc.

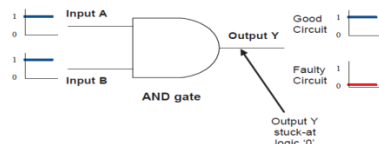


Fig 1.1: Behavior of stuck-at logic '0' fault at the output of an AND gate.

- *Transition delay fault model*

In order to model the delay defects in a digital circuit, the transition delay fault model was introduced. Like the stuck-at fault model, the transition delay fault model models faults on signal lines or interconnects between the gates. As per the transition delay fault model, a faulty signal line can behave as a slow-to-rise signal or a slow-to-fall signal. For a slow-to-rise transition delay fault on a faulty signal line, the signal line behaves as a temporary stuck-at logic '0' for a time period which exceeds the maximum delay of the circuit or is generally taken to be one test cycle or clock period. A similar behavior is exhibited by a slow-to-fall transition delay fault. Figure 1.2 shows the behavior of a slow-to-fall transition delay fault on the output of an AND gate. Transition delay faults model some of the physical defects that could arise in silicon manufacturing like gross delay defects in slow transistors, resistive shorting of signals to power lines, some cases of transistors permanently in OFF state, etc.

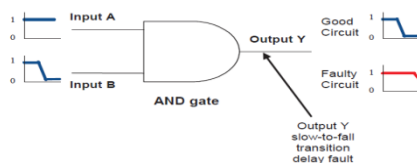


Fig 1.2: Behavior of a slow-to-fall transition delay fault at the output of an AND gate.

- *IDDQ faults*

In a CMOS gate, when the inputs of the gate are stable and not switching, then the current flowing between VDD and GND is negligible, ideally equal to zero. This steady state or quiescent current is termed as IDDQ current. However, in the presence of certain defects, it is observed that this current can increase by an order of magnitude as compared to the defect-free case. This observation enables detection of certain defects by measuring this current. Figure 1.3 shows an example of a short defect in a transistor in a NAND gate which causes abnormal IDDQ current to flow between VDD and GND. The IDDQ faults model certain physical defects occurring in fabrication like shorts between signal lines, transistors permanently in ON state, etc.

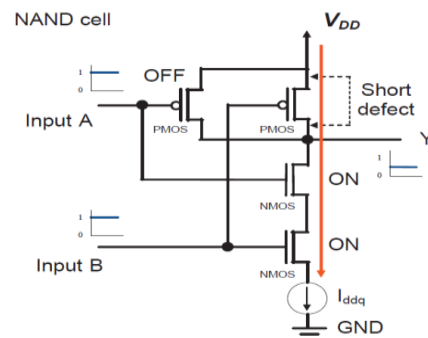


Fig 1.3: A short defect in a NAND gate causing abnormal IDDQ current between VDD and GND.

- *Bridging faults*

Other types of faults that have been defined at lower-levels of abstraction are bridging faults as shown in Figure 1.4, wire stuck-open faults, parametric faults, etc. Bridging Faults occur when signals are connected together.

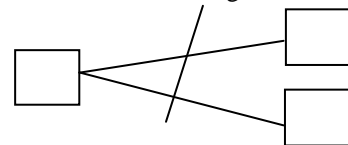


Fig 1.4: Bridging Fault.

C. Higher-level fault models

At higher levels of abstraction, faults can be modeled at the Register-Transfer Level (RTL) or the behavioral level. At the RTL, the digital design is modeled as data transfers between registers and faults can be modeled in the registers and/or in the data transfers between the registers. At the behavioral level, the digital design is described in the form of an algorithm or functional description and faults can be modeled in the various operations that are defined and used in the description. One of the fault models defined at higher abstraction levels like the RTL and behavioral level is the RTL fault model. Fault models at lower abstraction levels have higher correlation with the physical defects and hence are able to be characterized better as compared to fault models at higher levels of abstraction. However, fault models at higher abstraction levels are less complex and easier to analyze and utilize for test generation and test evaluation than those at lower abstraction levels. Hence, depending on the scenario, an appropriate fault model can be used.

2. FAULT INSERTION

A key issue in designing fault-tolerant digital systems is the validation of the design with respect to the dependability requirements. Fault injection is an effective method to study error behaviour, to measure the dependability parameters, and to evaluate fault-tolerant digital systems. Fault injection is the intentional insertion of faults into a system for the purpose of studying its response. Techniques for fault injection fall into two categories:

- (i) Simulation based fault injection, i.e. fault injection into simulation models of systems.
- (ii) Physical fault injection, i.e. fault injection into physical systems (prototypes or actual systems).

One advantage of simulation-based fault injection is that it can be used early in the design cycle. Therefore, design mistakes in the fault-tolerant system can be detected early and thus it reduces the cost for correcting such mistakes. It also provides a high degree of controllability and observability. However, the main drawback in simulation-based fault injection is, it is time-consuming especially when the simulated model is given in detail. One way to provide good controllability and observability as well as high speed in the fault injection experiments is to use FPGA-based fault injection. FPGA-based emulators have been used to inject stuck-at faults for test pattern generation purposes. The traditional FPGA-based fault injection methods cannot be used to inject faults into switch-level models as is more time-consuming than gate-level simulation (gate-level fault injection). This is because switch-level models are essentially more detailed than gate level models. For the purpose of injecting faults and validating the combinational circuit, a 4:1 multiplexer is considered in the gate level as shown in Figure 2.1. And the truth table for the same is given in table no 2.1.

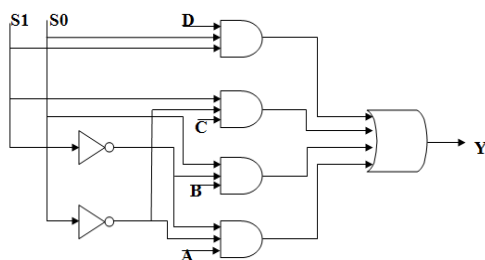


Fig 2.1: Logic Diagram of 4:1 MUX

Table 2.1: Truth table of 4:1 MUX

SELECT INES		OUTPUT
S1	S0	Y
0	0	A
0	1	B
1	0	C
1	1	D

A. Intentional insertion of faults into DUT

A 4:1 multiplexer is considered to be as a design under test (DUT) for the purpose of studying its response. The same 4:1 multiplexer is redesigned in-order to study the fault analysis

by inserting 2:1 multiplexer at each node of a 4:1 multiplexer, as shown in Figure 2.2.

The 2:1 multiplexer is used as a switch, where the select line of each 2:1 multiplexer is selected to insert a Stuck-at fault at the corresponding node of a 4:1 multiplexer, and one input of the 2:1 multiplexer is the output of previous gate and is passed to input of the next gate or node when select line is '0', and the other input of 2:1 multiplexer is used to inject either Stuck-at_0 fault or Stuck-at_1 fault when select line is '1'.

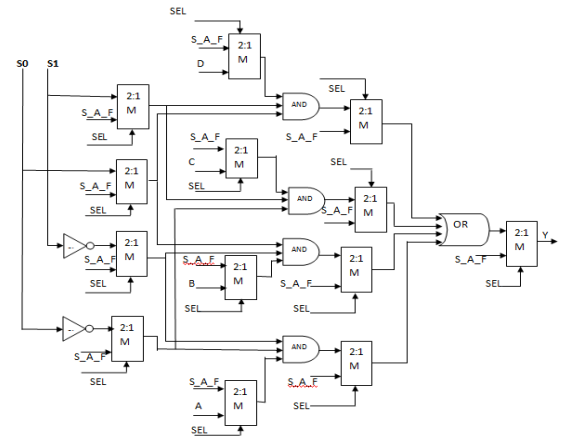


Fig 2.2: Re-Designed 4:1 MUX with fault injection capability.

3. THE D-ALGORITHM

The D-algorithm (D-ALG) is a search space comprising of all the internal nodes of the circuit along with the Primary Inputs (PIs) and is guaranteed to find a test vector for a fault if one exists. The D-algorithm has been specified very formally, and is suitable for computer implementation. It is the most widely used test vector generator. The primary aim of the D-algorithm is that, it always attempts to sensitize every possible path to the primary outputs. The two main components of the D- algorithm approach are the Automatic Test Equipment (ATE) and the output response analyzer.

A. Automatic Test Equipment (ATE)

Automatic or Automated Test Equipment (ATE) is any apparatus that performs tests on a device, known as DUT or unit under test, using automation to quickly perform measurements and evaluate the test results. An ATE can be simple computer controlled digital meter, or a complicated system containing dozens of complex test instrument capable of automatically testing and diagnosing faults in sophisticated electronic packaged parts or on wafer testing, including system-on-chip and integrated circuits.

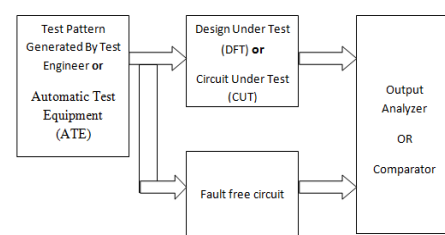


Fig 3.1: Block diagram of 4X1 MUX with fault injection capability.

B. Output Response Analyzer (ORA)

ORA produce qualitative reports to establish intervention goals. It is a comparative method, and Response Analyzer allows entering the data and automatically.

4. BUILT-IN SELF TEST (BIST)

Built-In Self Test (BIST) is a special case of Design-For-Test (DFT) methodology in which the circuit tests itself and flags whether it is good or bad. Additional hardware is inserted to generate test vectors which drive the primary inputs of the circuit, sample its primary output(s) and determine whether the circuit is good or bad by comparing the sampled output(s) with expected one(s). Random pattern testing has been a common practice in industry for a long time. It was found that it is quite easy to detect first, say 70% of the possible single stuck-at faults, and much more difficult to close up the gap up to 100%. So it is advantageous to utilize random test initial fault coverage, which is less costly to generate. In this project, the test pattern generator, part of BIST is implemented as an LFSR. The random test patterns are generated by the LFSR for combinational circuits.

The general block diagram of BIST architecture is as shown in Figure 5.1 the two main components of the BIST architecture are the test generator and the response analyzer. The test generator generates optimum test vectors to test the CUT such that high fault coverage is achieved. The response analyzer samples the primary outputs of the CUT, compares it with the expected good responses and flags whether the CUT is good or bad.

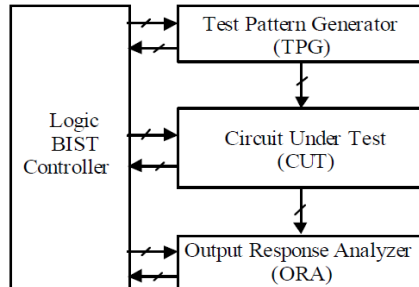


Fig 4.1: BIST system.

Along with these components, a BIST controller may be required to control the running of the BIST sessions. An efficient BIST architecture should be designed in such a way that it has low area overhead, high fault coverage and low test application time.

A. Test Pattern Generator

The random test patterns are generated by the LFSR for combinational circuits. One can build many kinds of LFSR but for BIST the most interesting kind is the so-called maximal length LFSR (ML-LFSR). An ML-LFSR enables to create a PRBS generator, which enables to generate almost all of the binary patterns.

The maximal length LFSR generates data that is almost random (hence the term 'pseudo-random'). The output of the LFSR can be taken in parallel-out form or as a serial bit

stream. The serial bit stream is usually taken from the MSB of the LFSR.

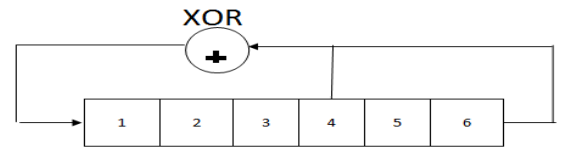


Fig 4.2: Maximal-length LFSR (6-Bit).

5. SIMULATION RESULTS

The simulation results for Normal 4:1 multiplexer is as shown in Figure 5.1, and the simulation result for 4:1 multiplexer which has a capability to insert faults, but no faults are inserted is as shown in Figure 5.2, and the simulation results for stuk_at_1 fault at I0 position is as shown in Figure 5.3.

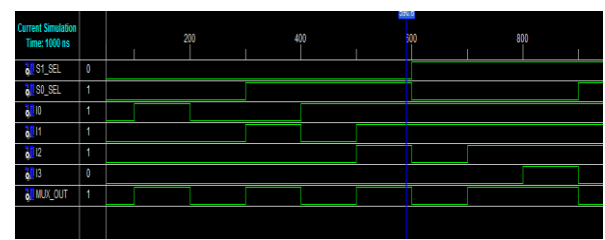


Fig 5.1: Simulation result for Normal 4X1 MUX.

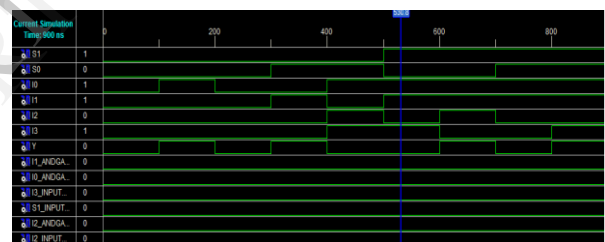


Fig 5.2: Simulation result for 4X1 MUX with Fault insertion capability (Without Fault insertion)

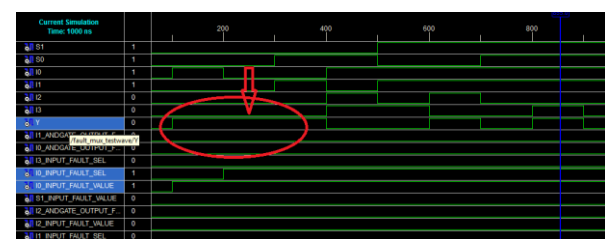


Fig 5.3: Simulation result for 4X1 MUX with Fault insertion capability (With Fault insertion)

A. D_Algorithm

The simulation result for 4:1 multiplexer by applying D-Algorithm without inserting the fault is as shown in Figure 5.4.

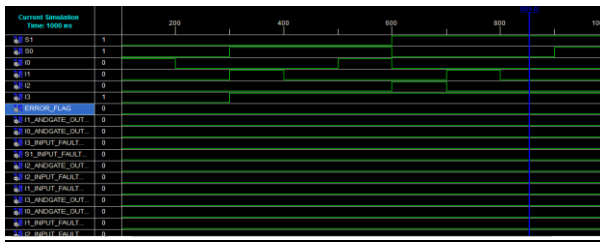


Fig 5.4: Simulation result for D_Algorithm (without Fault).

The simulation result for 4:1 multiplexer by applying D-Algorithm with stuck_at_1 fault at I0 is as shown in Figure 5.5.

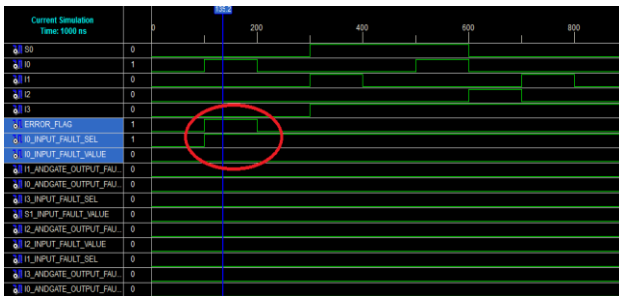


Fig 5.5: Simulation result for D_Algorithm (with Fault).

B. BIST

The simulation result of ATPG for generating test pattern for 4:1 multiplexer is as shown in Figure 5.6.

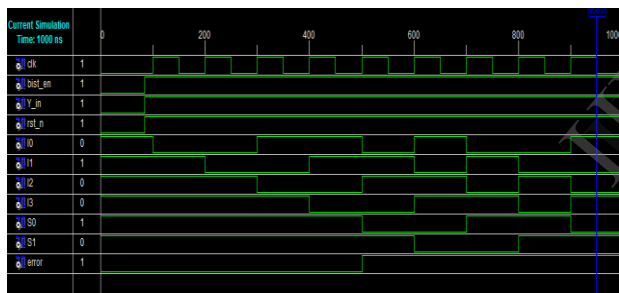


Fig 5.6: Simulation result for ATPG

The simulation result for 4:1 multiplexer with BIST Technique without inserting the fault is as shown in Figure 5.7.

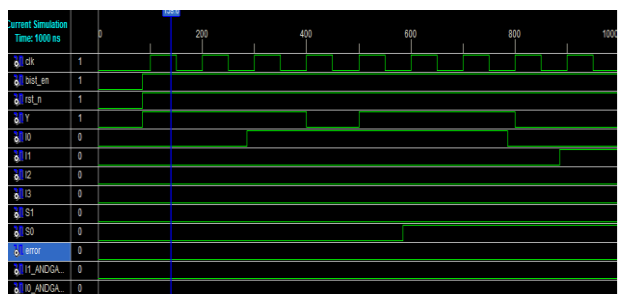


Fig 5.7: Simulation result for BIST (without Fault).

The simulation result for 4:1 multiplexer with BIST Technique with inserting the fault is as shown in Figure 5.8.

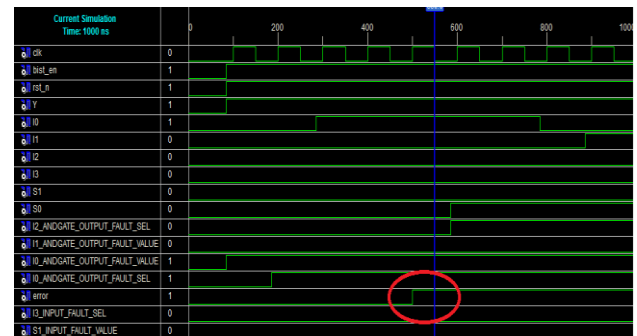


Fig 5.8: Simulation result for BIST (with Fault).

6. CONCLUSIONS

Validation and verification of the integrated circuits (IC's) is very much essential before it is manufactured. Due to increases in design complexity and shorter design cycles, design errors are more likely to escape detection and hence lead to high-cost field failure. Moreover, operational faults, which occur during normal operation, can also lead to high-cost field failure especially in high-availability and safety-critical applications. In this regard an attempt is made in realizing the occurrence of various faults in digital circuits such as single stuck-at faults and detecting the same. However, the fault analysis in these circuits is made by injecting the faults, which in turn validated by some techniques such as D-algorithm and BIST. In case of D-Algorithm, Automatic Test Equipment is used to generate test vector and in case of BIST, maximal length LFSR (ML-LFSR) is used to generate a test vector, which generate all of the required binary patterns for the circuit under test. BIST eliminates the need for ATPG, which in general used for generating test patterns and analyze the responses. BIST is a Design For Testability (DFT) technique, in which, an additional hardware and software features are incorporated into integrated circuits to allow them to perform Self-Testing.

ACKNOWLEDGMENT

The authors thank the faculty members of E&CE Dept. and authorities of Sri Dharmasthala Manjunatheshwara College of Engineering and Technology, Dhavalagiri, Dharwad, Karnataka, India for encouraging us to carry out this research work.

REFERENCES

- [1] Zvi Kohavi, "Switching and Finite Automata Theory", Third Edition, Cambridge university press Publications.
- [2] Giovanni De Micheli, "Synthesis and optimization of Digital Circuits".
- [3] Srikanth Alaparthi, Prasanth Jampani, "Study and design of various bist approaches for combinational circuits", Dept. of Computer Engineering, Texas A&M University, College Station, TX – 77840.
- [4] Nitin Yogi, "Spectral Methods for Testing of Digital Circuits", Electrical and Computer Engineering, Auburn, Alabama, August 10, 2009.
- [5] Alireza Ejlali, Seyed Ghassem Miremadi "FPGA-based fault injection into switch-level models", Elsevier, 3 march 2004.
- [6] Ramesh Bhaktavachalu, Deepthy G R, "32-bit Reconfigurable Logic-BIST Design Using Verilog for ASIC Chips", IEEE, 2011.
- [7] Graham Hetherington, Tony Fryars, NageshTamarapalli, Mark Kassab, Abu Hassan, and JanuszRajski, "Logic BIST for Large Industrial

Designs: Real Issues and Case Studies”, proceedings of the international test conference, 1999, page 358 – 367.

- [8] Laung-Terng Wang, Cheng-Wen Wu, Xiaoqing Wen, “VLSI test principles and architectures design for testability”, Morgan Kaufmann Publishers, Elsevier.
- [9] SM.Thamarai , Dr K.Kuppusamy, Dr T.Meyyappan, Fault Detection and Test Minimization Methods for Combinational Circuits – ASurvey.ISSN:2231-2803.
- [10] Sachin Dhingra, ELEC 7250: VLSI testing, Implementation of an ATPG using PODEM algorithm.
- [11] Hussain Said Al-Asaad, “Lifetime validation of digital systems via fault modeling and test generation”, Dept. of Computer Science and Engineering, the University of Michigan 1998.

Authors



Mr. Umesh is pursuing his 2nd year Master of Technology in Digital Electronics at SDM College of Engineering & Technology, Dharwad under Visvesvaraya Technological University, Belgaum.

He received Bachelor of Engineering in Electronics and Communication Engineering from Sridevi Institute of Engineering and Technology, Tumkur, under Visvesvaraya Technological University, Belgaum in 2011. His research interests include Digital Circuits and application, Embedded system, Design for Testability, CMOS VLSI, ASIC Design.



Mr. Kotresh E. Marali received Bachelor of Engineering in Electronics and Communication Engineering from Visvesvaraya Technological University, Belgaum in 2009 and Master of Technology in Digital Electronics under Visvesvaraya

Technological University, Belgaum. Currently he is working as Asst. Professor in the Department of Electronics & Communication Engineering at SDM College of Engineering & Technology, Dharwad. His main research interests include VLSI design and Computer Architecture, and ASIC Design.