LFSR TEST PATTERN FOR FAULT DETECTION AND DIAGNOSIS FOR FPGA CLB CELLS

Fazal Noorbasha, K. Harikishore, Ch. Hemanth, A. Sivasairam, V. Vijaya Raju
Department of ECE, KL University, Vaddeswaram, Guntur (Dist.), AP, India

ABSTRACT
The increasing growth of sub-micron technology has resulted in the difficulty of VLSI testing. Test and design for testability are recognized today as critical to a successful design. Field Programmable Gate Arrays (FPGAs) have been used in many areas of digital design. Because FPGAs are reprogrammable, faults can be easily tolerated once fault sites are located. In this paper, we discuss about fault detection and fault diagnosis techniques for FPGA CLBs. The most of the discussion will be made using Configurable Logic Block (CLB) instead of whole FPGA for simplicity. In order to generate testing pattern we aid Linear Feedback Shift Register (LFSR). Fault detection and location will be carried out using output response analyzer. Diagnosis will be discussed accordingly. For this analysis we have taken XC4000 FPGA. VHDL is used as HDL language.

KEYWORDS: *FPGA, CLB, LFSR, XC4000, VHDL*

I. INTRODUCTION

Field programmable gate arrays (FPGAs) are a popular choice among VLSI devices, any logical circuit can be implemented into the FPGA at low cost. It consists of an array of configurable logic blocks (CLBs), programmable interconnect and programmable Input/output blocks. Many methods have been proposed to test FPGAs [1]. In some works, the circuits under consideration are programmed FPGAs, in which logic circuits have been implemented. Since an FPGA can be programmed in many different ways, this method is not applicable to manufacturing time testing, as we do not know the final configuration. Testing faults in general FPGAs has been proposed by many researchers. In these methods, the FPGA under test is not mapped to a specific logic function. As a result, multiple test sessions are usually required, with each session dealing with one configuration. The BIST architecture requires the addition of three hardware blocks to a digital circuit: a pattern generator, a response analyzer and a test controller. Examples of pattern generators are a ROM with stored patterns, a counter and a linear feedback shift register.

LFSR is constructed using flip-flops connected as a shift register with feedback paths that are linearly related using XOR gates. An LFSR can be used for generation of pseudo-random patterns, polynomial division, response compaction etc. [2]. LFSR is more popular for implementation of both TPG and ORA due its compact and simple structure. Figure 1 shows the generic FPGA architecture for VLSI design.

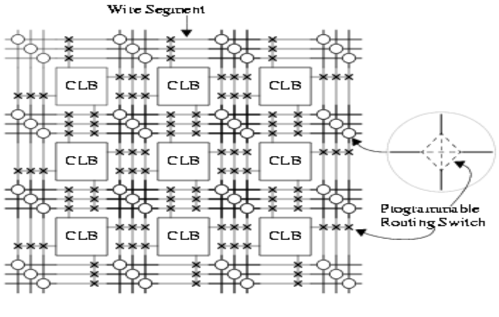

Figure 1. Generic FPGA architecture

A typical response analyzer is a comparator with stored responses or an LFSR used as a signature analyzer. Traditional chip testing deals with fault detection only, while fault diagnosis is often conducted at the system level. This is because components in the chip cannot be repaired. However, faults in FPGAs can be easily tolerated by not including faulty elements in the final circuit. Therefore, FPGA chips with faults can still be used if we can identify the fault sites. In this paper, we propose a chip-level diagnosis methodology for faults in CLBs. Our method is also based on the BIST technique, which means that the testing process is conducted by the chip itself, and the requirement for external ATE support is limited.

II. Preliminaries and Fault Diagnosis

In order to diagnose faults, there must first be a way to test modules in FPGAs. A candidate for this purpose is the built-in self-test (BIST). This structure reconfigures part of the functional circuit to be a test pattern generator (TPG), and some other to be an output response analyzer (ORA) [4]. The rest circuit consists of the circuit under test (CUT). The TPG is either a linear feedback shift register (LFSR) that generates pseudorandom test sequences, or simply a counter that provides an exhaustive test set.

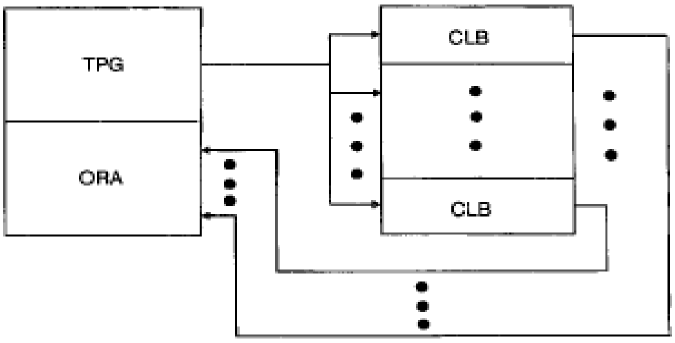

Figure 2. Connections between testing module and set of CLBs under test

The test inputs are fed to the CUT, while the output responses are collected and analyzed by the ORA. The ORA can be either a signature analyzer or a comparator-based analyzer [5]. Our test architecture works as follows. An FPGA is divided into disjoint sets of CLBs, where each set can be configured into a TPG and ORA as shown in figure 2. Such a set acts as a module in the PMC model since it is able to test another module and determine whether the CUT passes or fails the given test. All the CLBs under test are programmed in the same way; therefore, they perform the same logic function and could be given the same test patterns. Thus outputs of the TPG are fed to all CLBs in the set under test, and the results are analyzed by the ORA. Since each CLB can be programmed in many ways, it is not possible to test any CLB in a single test run. As a result, a complete test of all faults in a CLB usually requires several steps, and in each step a CLB is programmed in a particular way.

Our fault diagnosis process contains two sessions: 1) horizontal diagnosis, and 2) vertical diagnosis. Horizontal diagnosis is illustrated in figure 3 (a). The outputs of the rightmost cells are compared with correct results for identifying faulty rows.

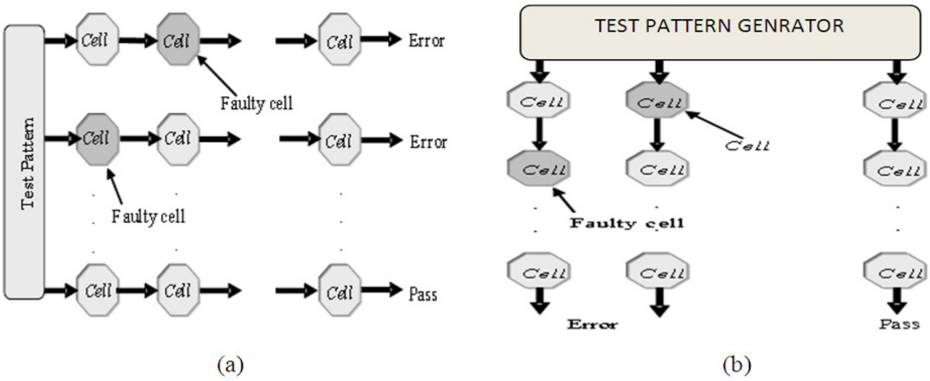

Figure 3. (a) Horizontal Diagnosis (b) Vertical diagnosis

The interconnection structures for vertical diagnosis are shown in figure 3(b). The outputs of the bottommost cells are compared with correct results for identifying faulty columns. By intersecting faulty columns with faulty rows, the faulty cells can be identified. The diagnosis resolution of our approach is a single CLB due to the structure of a basic cell.

We arranged the CLBs shown above in an array of 8×8 for the experiment purpose. Since CLBs are configurable; we can 'inject' faults into CLBs by simulating the effects of logic faults. We consider two types of faults: stuck-at faults and open faults. We simulate the effect of a stuck-at fault by connecting the faulty line to logic '0' or '1'. To simulate the effect of an open fault, all we have to do is to disconnect the faulty line from its driving source [6]. Once by using the above diagnosis we can find the faulty CLBs easily. Once the location is found we can avoid those CLBs from programming and FPGA still be used efficiently.

III. LFSR FOR TEST PATTERN GENERATION

Different structures of LFSR will generate different sequence of test pattern. It means that if the BIST time is limited, the structure of LFSR will affect the BIST time and Fault Coverage (FC) of Circuit under Test (CUT). LFSRs are more popular because of their compact and simple design. Cellular Automaton LFSR are more complex to design but provide patterns with higher randomness and perform better in detection of faults such as stuck-open or delay faults, which need two-pattern testing. A software tool is used which automatically generates built-in self-test blocks into VHDL models of digital circuits by giving the suitable values of initial seed and primitive polynomial in TPG block [3]. Figure 4 shows the LFSR for test pattern of G and F of XC4000 FPGA. It will generate 8-bits, 4-bit for G and 4-bit for F.

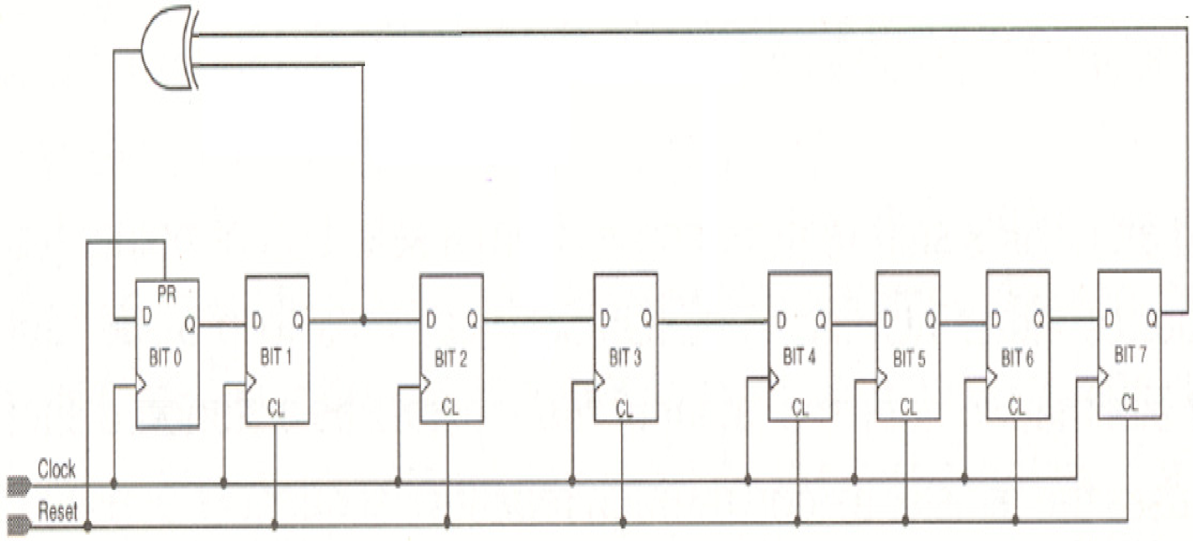

Figure 4. LFSR test pattern signals generator of G and F inputs of XC4000 FPGA

IV. TESTING OF FPGA CLB

A simplified diagram of the CLB in Xilinx XC4000 Family is shown in figure 5. This CLB has 13 inputs *(I*= 13), in which one is the clock input (K), nine signals are input to the combinational part (F1 to F4, G1 to G4, and Hl), and the other three are for the sequential part (DIN, S/R, and EC). There are four outputs in a CLB (0 = 4), in which two are combinational outputs and the other two are for sequential circuits. The combinational part consists of three look-up tables (LUTs), and three multiplexers (MUXs), whose outputs are H1, X, and Y, respectively. The sequential part is made up of the remaining components: two D flip-flops *(F*=2)**,** the S/R control, and the remaining MUXs.

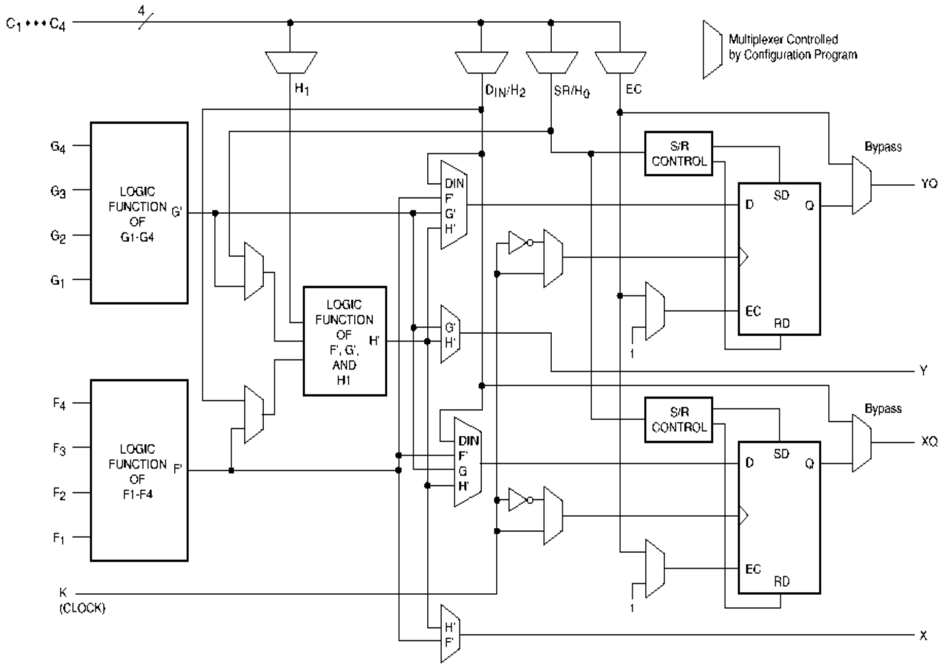

Figure 5. Simplified diagram of a CLB in XC4000

The testing of FPGAs falls into two categories: testing of unprogrammed FPGAs (configuration-independent testing) and testing of programmed FPGA (configuration-dependent testing) [7]. Here, we focus on the testing of unprogrammed FPGAs. The LUTs consist of SRAM. To test the memory elements, each bit has to be set to both '1' and '0'. Therefore, at least two phases are required to exercise all possible faults in the LUTs. There are 4-to-1 MUXs in a CLB. As result, we need at least four test phases so that each input-output connection of these MUXs can be exercised. We found that four test phases are enough to exercise all possible configurations in the CLB. These configurations are given as follows.

Phase I: The LUTs are configured as exclusive-OR (XOR) of the nine inputs. F' is connected to both flip flops, G' is connected to Y and H' is connected to X.

Phase 2: The LUTs are configured as exclusive-NOR (XNOR) of the nine inputs. G' is connected to both flip flops, F' is connected to X, and H's is connected to Y.

Phase 3: The LUTs implement XOR, and DIN is connected to both flip-flops.

Phase 4: The LUTs implement XNOR and H'is connected to both flip-flops.

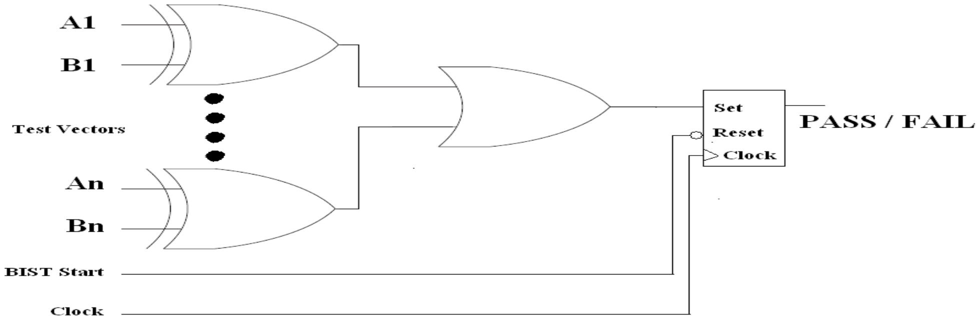

Figure 6. Comparator based ORA

The combinational part is tested in the first two phases. Flip-flops are tested in all four phases. In order to fully test the upper MUXs, in phase 1 the four connections are (C 1, C2, C3, C4)=>(Hl, DIN, S/R, EC) (i.e., H1 is connected to C1, DIN is connected to C2, etc.). In phase 2, the connection is (C2, C3, C4, Cl) => (Hl, DIN, S/R, EC), and in phase 3 we have (C3, C4, C1, C2) => (Hl, DIN, S/R, EC). Finally, in phase 4 the connection is (C4, C1, C2, C3) => (Hl, DIN, S/R, EC). As a result, all connections are exercised in four phases. The testing pattern is generated using 8-bit LFSR whose polynomial is $x^8 + x^6 + x^5 + x + 1$. The output response analyzer we used here is a comparator based

ORA. There are *B* CLBs in a node, and each CLB has four output lines. Since all C-mode CLBs are configured in the same way, they should give the same output response if all CLBs are fault-free [8]. Thus the same output signals in all CLBs are fed to one ORA for comparison as shown in figure 6(e.g., output YQs in all C-mode CLBs are sent to the same ORA). We need four ORAs as each CLB has four output lines. In each ORA, all *B* inputs are fed to a XOR gate. If all CLBs under test are fault-free, the output of the XOR gate should always be zero. Unless the number of faulty CLBs is even and those faults are identical, the XOR gate will generate a 1 output at least once, and that will set the FF to be 1 and the detected fault is recorded.

V. Simulation Results

The following results are obtained using XILINX tool. The Xilinx XC4000 model is used for fault modeling. Initially we designed front end model. The function generators are programmed with simple functions .For testing purpose we generated TPG using 8-bit LFSR. The test pattern results S/R control is shown in below figure 7. The LFSR G and F test patterns are shown in below figure 8. The outputs of LFSR are fed to the CLB function generators. LFSR used here is of external type. The LFSR generates 8-bits using this 8-bits we have taken two 4-bits for G and F to test the FPGA CLBs. Initially it was implemented in DSCH and later in Xilinx for better view of results. Faults are incorporated into the CLBs for testing purpose. Faults that are incorporated here are stuck-at fault and open fault. By using this test pattern we have tested only stuck at faults and open faults.

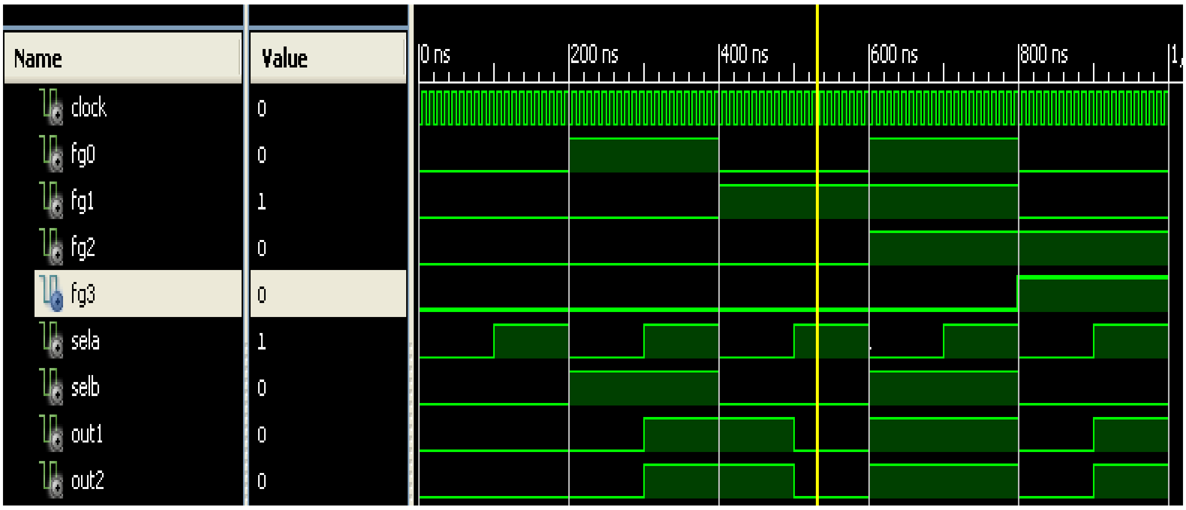

Figure 7. S/L control signals for XC4000 FPGA CLBs

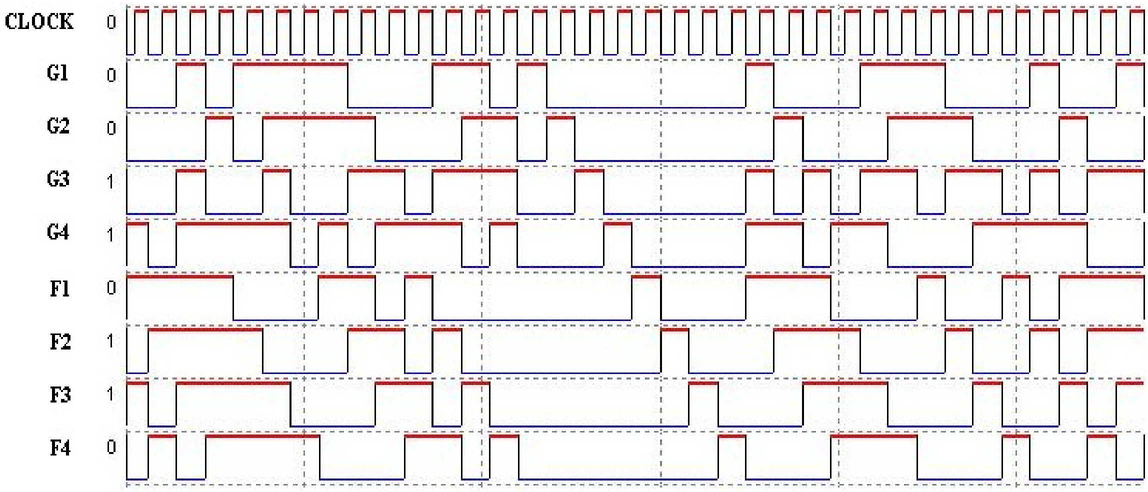

Figure 8. G and F input test pattern signals of XC4000 FPGA generated by 8-bit LFSR

VI. CONCLUSION

In this paper we present a methodology for the diagnosis of faulty CLBs in FPGAs. The reference CLB we have used is XC4000 series. For generation of test pattern 8-bit LFSR is used. Analysis purpose we have injected some of the common faults like stuck at fault and open fault. In this method all the testing is done within the FPGA which prevents the use of external hardware. The main advantage of this method is testing time of the CLBs mainly depends on number of faults rather than the chip size. Hence it yields more advantages while diagnosing the larger chips. For testing of delay faults and stuck-open faults, test patterns which are more random in nature provide better fault coverage, so in many cases they are being looked upon as the alternatives to conventional LFSRs for test pattern generation and output response analysis in BIST. Also the resolution of the fault diagnosis algorithm in interconnect testing is greatly dependent on the structure of the original application configuration. The complexity of the CLB test configuration will increase in some worst case. All these problems need to be studied in the future research work in this area.

REFERENCES

[1]. D. Gil, J. Gracia, J.C. Baraza, P.J. Gil, (2003) "Study, Comparison and Application of different VHDL-Based Fault Injection Techniques for the Experimental Validation of a Fault-Tolerant System", Microelectronics Journal, vol. 34, no. 1, pp. 41-51.

[2]. Guihai Yan,E, Yinhe Han, And Xiaowei Li, (2011) "SVFD: A Versatile Online Fault Detection Scheme via Checking Of Stability Violation", IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol.19, No.9, pp.1627-1640.

[3]. Shikha Kakar, Balwinder Singh and Arun Khosla, (2009) "Implementation of BIST Capability using LFSR Techniques in UART", International Journal of Recent Trends in Engineering ,Vol 1, No. 3, pp. 301-304.

[4]. Mehdi B. Tahoori,(2011) "High Resolution Application Specific Fault Diagnosis of FPGA", IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 19, No. 10, PP.1775-1786.

[5]. I. Pomeranz and S. Reddy,(2004) "Improving the stuck-at fault coverage of functional test sequences by using limited-scan operations," IEEE Transactions On Very Large Scale Integration (VLSI) Systems, vol. 12, no. 7, pp.780–788.

[6]. P. Bernardi,(2006) "A pattern ordering algorithm for reducing the size of fault dictionaries," in Proc. IEEE VLSI Test Symp., pp. 45–51.

[7]. N. Hernandez and V. Champac, (2008) "Testing skew and logic faults in SOC interconnects," in Proc. IEEE Comput. Soc. Annu. Symp. VLSI, pp. 151–156.

[8]. A. Namazi, M. Nourani, and M. Saquib, (2007) "A robust interconnect mechanism for nanometer VLSI," presented at the Int. Test Synthesis Workshop (ITSW), San Antonio, TX.

Authors

Fazal Noorbasha was born on 29th April 1982. He received his, B.Sc. Degree in Electronics Sciences from BCAS College, Bapatla, Guntur, A.P., Affiliated to the Acharya Nagarjuna University, Guntur, Andhra Pradesh, India, in 2003, M.Sc. Degree in Electronics Sciences from the Dr. HariSingh Gour University, Sagar, Madhya Pradesh, India, in 2006, M.Tech. Degree in VLSI Technology, from the North Maharashtra University, Jalgaon, Maharashtra, INDIA in 2008, and Ph.D. Degree in VLSI from Department Of Physics and Electronics, Dr. HariSingh Gour Central University, Sagar, Madhya Pradesh, India, in 2011. Presently he is working as a Assistant Professor, Department of Electronics and Communication Engineering, KL University, Guntur, Andhra Pradesh, India, where he has been engaged in teaching, research and development of Low-power, High-speed CMOS VLSI SoC, Memory Processors LSI's, Fault Testing in VLSI, Embedded Systems and Nanotechnology. He is a Scientific and Technical Committee & Editorial Review Board Member in Engineering and Applied Sciences of World Academy of Science Engineering and Technology (WASET), Advisory Board Member of International Journal of Advances Engineering & Technology (IJAET), Member of International Association of Engineers (IAENG) and Senior Member of International Association of Computer Science and Information Technology (IACSIT). He has published over 20 Science and Technical papers in various International and National reputed journals and conferences.

Harikishore Kakarla was born in Vijayawada, Krishna (Dist.), AP, India. He received B.Tech. in Electronics & Communication Engineering from, JNTU, Hyderabad, AP, India, M.Tech. from SKD University, Anantapur, AP, India. He is pursuing Ph.D in the area of VLSI in KL University, Vijayawada, AP, India. He has published 03 International Journals and 01 National Conference Level. Presently he is working as a Assistant Professor, Department of Electronics and Communication Engineering, KL University, Guntur, Andhra Pradesh, India, where he has been engaged in teaching, research and development of Low-power, High-speed CMOS VLSI SoC, Memory Processors LSI's, Fault Testing in VLSI, Embedded Systems and Nanotechnology.