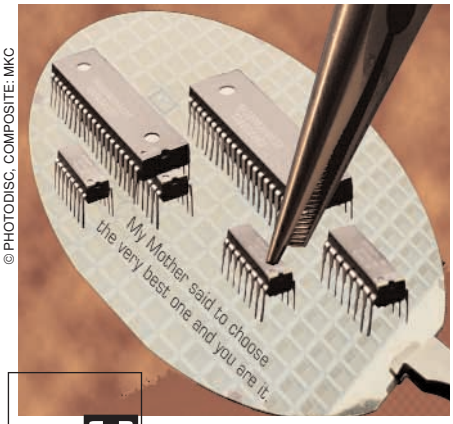


the state of VLSI testing

L.M. PATNAIK, H. S. JAMADAGNI, V.K. AGRAWAL AND B.K.S.V.L.VARAPRASAD

in trying to keep costs down specialized testers are now all the rage

The phenomenal development in electronic systems has, in large part, the advances in Very Large Scale of Integration (VLSI) semiconductor technologies to thank. Performance, area, power and testing are some of the most important improvements. With the reduction in device sizes, it is becoming possible to fit increasingly larger number of transistors onto a single chip. However, as chip density increases, the probability of defects occurring in a chip increases as well. Thus, the quality, reliability and cost of the product are directly related to the intensity/level of testing of the product. As a result, gradually, Integrated Circuits (ICs) testing has shifted from the final fabricated ICs to the design stage. In this context, many Design For Testability (DFT) techniques have been developed to ease the testing process.

Test methods

The functionality of a combinational circuit can be verified by exhaustively applying all possible input patterns to its inputs. Testing involves generating test patterns, applying them to the circuit then analyzing the output response. Testing falls into a number of categories depending upon the intended goal.

The *functional* test (also called the go/no go test) determines whether or not a manufactured component is functional. Since this test gets executed on every manufactured die and has a direct impact on the cost (viz., test equipment cost, testing time), it should be as simple and swift as possible.

The *parametric* test checks on a number of parameters, such as voltage levels, current levels, noise margin, propagation delays, capacitive coupling or cross talk and maximum clock frequencies. These parameters are tested under a variety of working conditions, such as temperature and supply voltage.

As circuits get faster and more complex, the measurement of "transient" characteristics governing their timing-dependent behavior becomes increasingly important. "AC" (high frequency and pulse) testing or at-speed testing simulates circuits up to their highest projected operational speeds or frequencies then observes the "behavior."

Path-delay or *transition-delay* techniques are extensions of stuck-at faults (i.e., structural) testing. They have been developed to catch timing related defects. With shrinking geometry and thinner gate dielectric, bridging between different tracks and gate-oxide shorts have become the most common defects in Complementary Metal Oxide Semiconductor (CMOS) technology. Only I_{DDQ} *testing* can detect these defects. The I_{DDQ} test measures the supply current of a device while the device is on a quiescent state. A good estimate of the supply current in the fault-free circuit is essential for I_{DDQ} testing.

Substrate leakage current is very high in Deep Sub Micron process technologies. Also, there is no known mechanism that provides a good estimate of fault-free current in a reasonably acceptable time. To combat this problem, a *differential current* testing technique is proposed, where a differential current measurement taken at a given input (ΔI_{DDQ}) is defined as the I_{DDQ} measurement taken at this input minus the I_{DDQ} measurement taken at the previous input. Finally, a *diagnostic* test is used during the debugging of a chip or board and tries to identify and locate the fault in a failing chip or board.

Fault models and fault simulation

Traditionally, physical circuit defects are often modeled as logic faults. The assumption being that any manufacturing defect will translate itself into an erroneous logic value. This assumption is called fault modeling. This makes the problem of fault analysis independent of the technology.

Fault modeling analyzes the circuit's behavior in the presence of faults caused by physical defects or environmental influences. Hence, fault modeling provides a basis for the fault simulation and test generation.

Test pattern validation is the process of determining how well a test pattern meets the fault coverage requirements. Test coverage/fault coverage is defined as the ratio of the faults that would be detected and the total number of faults in the assumed fault universe. This is a very important step in the testing process. It provides a measure of adequacy for a given test set.

In Deep Sub Micron process technology, delay faults and cross-talk faults are becoming dominant faults in System-on-Chip (SoC) designs. The testing of the interconnects requires modeling cross-talk faults in addition to

0278-6648/02/$17.00 © 2002 IEEE

stuck-at faults and bridging faults.

Cross talk can occur when there is capacitive coupling between a pair of traces and the cross talk hazard can be modeled by a lumped capacitance. Signals on two lines that run in proximity can influence one another. For example, a 0-1 (i.e. logic 0 to logic 1) transition on one line (known as the aggressor) can cause a spurious 0-1-0 glitch on the other line (known as the victim line). This cross-talk fault will be detected only if clocking occurs during the glitch because it is intermittent.

Power supply noise, is a result of switching activity in the circuit. The power supply noise reduces the actual voltage level reaching a device, which in turn increases propagation delay of gates. The maximum allowable clock rate in a digital circuit is determined by the propagation delays of the combinational logic network between the latches. If the delay of the combinational network exceeds its specified value then incorrect logic values may be latched in flop-flops. Delay faults can be modeled into two different fault models viz., the single-gate delay fault model and the path-oriented delay fault model. The path delay fault model has gained importance compared to the gate delay model, because it models excessive delays that are local as well as distributed. In path delay fault model, any path with a total delay exceeding the system clock interval is said to have a path delay fault. This is a distributed fault model because it is associated with an entire path. The major bottleneck in the path delay fault model is the selection of paths for which test generation and fault simulation are to be carried out since, as the circuit size grows, the number of paths grow exponentially with circuit depth and the number of fan outs.

Test generation schemes

The functionality of a combinational circuit can be verified by exhaustively applying all possible input patterns to its inputs. For an N-input circuit, this requires the application of 2^N patterns. In a sequential circuit, the output of the circuit depends not only on the inputs applied, but also on the state of the internal registers of the circuit. An exhaustive test of a sequential circuit requires the application of 2^{N+M} test patterns, where N and M are the number of inputs and registers/flip-flops of the circuit respectively. As N and M increase, the test length expands exponentially. To reduce the test pattern length, an alternate approach is required.

Test generation is the process of determining a set of stimuli necessary to test a circuit. The computation cost of the test generation depends on the complexity of the method. A test for a fault can be found by trying various input patterns until one gives a different output so that the considered fault is found in the circuit.

The generation of tests for combinational digital circuits can be characterized as a search problem. Finding an efficient way to try these different patterns makes the problem amenable to the use of search techniques. The search space can be quite large and a solution may not exist. Some method must be used to organize the search process to assure that every possible pattern can be tried and that no pattern is tried twice. It is not necessary, however, to explicitly search the entire space; it may be possible in some cases to determine that no solution exists with only a partial search. Similarly, if a way can be found to search a smaller space for the solution, the size of the search problem is reduced and the test generation problem becomes more tractable.

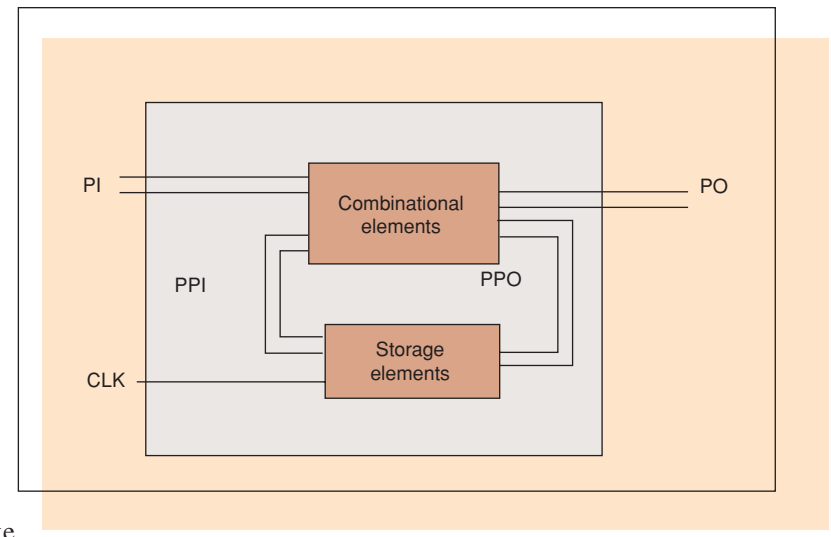


Fig. 1 Mealy synchronous sequential circuit

Random test generation (RTG) is a simple process that involves only the generation of random vectors. However, to achieve a high-quality test we need a large set of random vectors. RTG works without taking into account the function or the structure of the circuit to be tested.

In contrast, deterministic test generation produces tests by processing a model of the circuit. Deterministic test generation can be fault-independent or fault oriented. Fault-independent test generation works without targeting individual faults. One early approach in test generation was using the concept of boolean difference for generating the test patterns. The boolean difference approach can be characterized as algebraic. It manipulates circuit equations to generate test patterns.

In the context of VLSI systems, test generation by algebraic methods is too time-consuming; hence, the algorithmic approaches are being used. In fault-oriented process, tests are generated for specified faults of a fault universe. Some of the most well known test generation algorithms that have been developed are D-algorithm, Path-Oriented Decision Making (PODEM), Fanout-Oriented (FAN) and Topological Search (TOPS). These test generation methods use the topological gate level description of the circuit.

The algorithmic approaches also use various mechanisms to trace sensitive paths to propagate fault effects to primary outputs. They then back trace to the primary inputs and assign logic values based on conditions set at the forward fault propagation stage. The test generation methods, i.e., D-algorithm, PODEM, FAN and TOPS, generate test vectors for one fault at a time. But as the number of internal nodes in Circuit Under Test (CUT) increases, the computation time in generating the input test patterns becomes enormous.

Thus, in VLSI systems, the time required to generate the test patterns for the whole system to get full fault coverage is quite high. A common way to reduce the test generation cost is the "divide and conquer" method: wherein the problem is partitioned into smaller parts, the solution to the parts is obtained and then combined into a solu-

tion for the whole problem.

By using the deterministic and random pattern testing in combination, the test system can be made more open-ended and extendable. For example, initially a sequence of random pattern, at-speed tests can be used to detect errors. This will catch some 80% or 85% of all possible faults. Then, deterministic tests (which employ aimed patterns, and can be exhaustive when necessary) can be used to deal with the faults not caught by the random patterns.

The traditional Automatic Test Pattern Generation (ATPG) methods like PODEM and FAN target the test generation problem at the logic level. However, they can require large amounts of computing time and resources to generate tests for even moderately sized sequential circuits. Computation time can be drastically reduced with mapping the Register Transfer Level (RTL) signals of the circuit being tested to its corresponding gate level nets. Several methods have been reported which automatically use functional RTL descriptions of the circuit that target detection of stuck-at-faults in the circuit at the logic level.

Design For Testability (DFT)

The challenge of designing for testability stems from the fact that an integrated circuit has many more internal nodes than input or output pins. Hence, the value of an internal node has to be forced to a given value or even knowing the value of an internal node may be extremely difficult if not impossible.

When considering test designs, a circuit's testability depends on the controllability and observability of the circuit's internal nodes. Controllability measures the ease of bringing a circuit node to a given value/state using only the input pins. A node is easily controllable if it can be brought to any condition by only a single input vector. A node (or circuit) with low controllability needs a long sequence of vectors to be brought to a desired state.

Observability measures the ease of observing the value of a node at the output pins. A node with a high observability can be monitored directly on the output pins. A node with a low observability needs a number of cycles before its state appears on the outputs. Combinational circuits typically fall under the class of easily observable and controllable circuits, since any node can be controlled and observed in a single cycle.

In sequential circuits, the fault detection process is much more complex. Indeed, the presence of memory elements and feedback lines adds further difficulties in bringing the sequential circuit into a required state. Several test patterns may be needed to detect a fault.

For example, the detection of each fault in the Mealy model of a synchronous sequential circuit as shown in Fig. 1 implies:

1. Setting Primary Inputs (PIs) and Pseudo-Primary Inputs (PPIs) to specific values;
2. Propagating the fault effect to the Primary Outputs (POs).

Setting the PPIs is not a straightforward operation. Indeed, the PIs do not directly control the PPIs. For instance, if the assignments needed at PPIs to test the logic correspond to a non-valid state, the fault is impossible to detect without some form of built-in self-test (BIST). Moreover, observing the effect of a fault directly at the POs is also not obvious. Indeed, the effect may only propagate to the Pseudo-Primary Outputs (PPOs). Thus, we may need additional cycles to propagate to the POs. If it happens that the effect of a fault brings the faulty circuit in an equivalent state to that of the fault-free circuit, then the fault is impossible to detect.

Design For Testability (DFT) is the design effort through which a circuit can be made easily testable by enhancing the controllability and observability. The DFT techniques are divided into two categories. The ad hoc techniques solve a problem for a given design and are not generally applicable to all designs. The structured techniques are generally applicable and usually involve a set of design rules by which designs are implemented. The objective of the structured approach is to reduce the sequential complexity of a network to aid test generation and test verification.

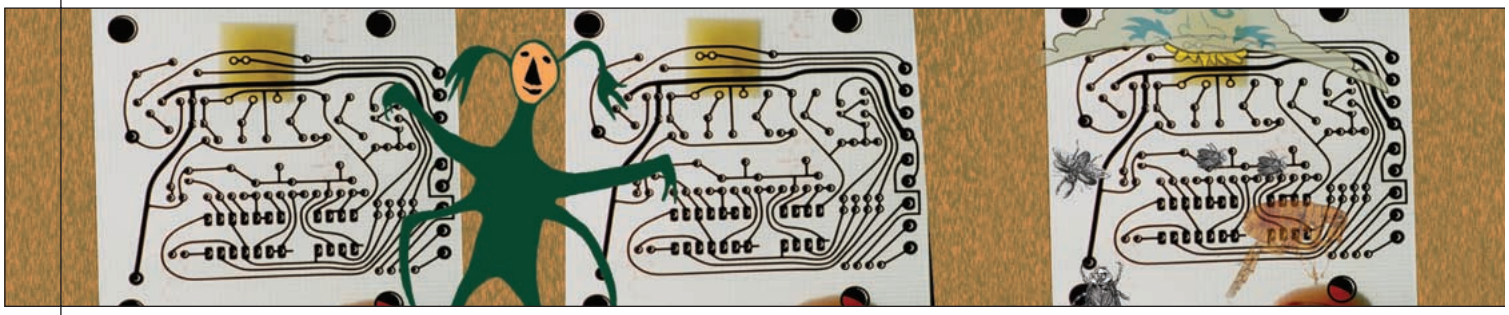
Ad hoc test techniques: Certain techniques are considered to be ad hoc because they do not provide an efficient test solution. Also, they can be used at the designer's option where applicable. Some of the commonly used ad hoc techniques are:

• Employing *test points*, viz., control points and observation points, to enhance controllability and observability of the internal nodes makes the circuits fully testable. Control points are primary inputs used to enhance controllability and observation points are primary outputs to enhance observability of a CUT.

• *Initialization* is the process of bringing a sequential circuit into a known state at some known time, such as when it is powered on or after an initialization sequence is applied. For many designs, initialization can most easily be accomplished by using the asynchronous preset (PR) or clear (CLR) inputs to the flip-flops.

• Counters and to a lesser extent shift registers, are difficult to test because their test sequences usually require many clock cycles. Hence *partitioning large counters and shift registers into smaller units* increases testability and reduces test time.

Structured methods: The most popu-



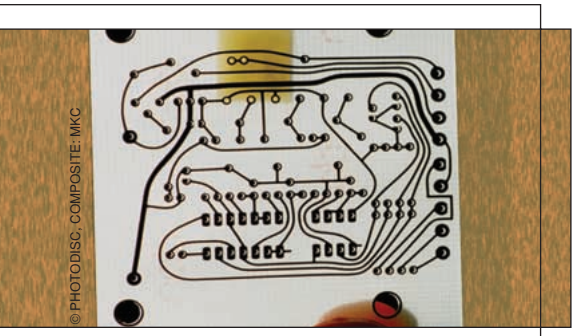
One of these is not like the others

lar structured DFT technique used for testing is referred to as scan design since it employs a scan register.

Full Serial Integrated Scan: A sequential circuit, S, can be converted into full serial integrated scan or scan path circuit. All the flip-flops of the sequential circuit are changed by flip-flops that accept two input signals D_1 and D_2 depending on the flip-flop control input T. Here, D_1 is used for the data input during the normal mode of operation when T = 0. D_2 is used for testing the input when T = 1. The D_2 input is used to connect the flip-flops so they form a shift register.

The sequential part of the circuit **S** is now easily controllable and observable because values to the registers can be shifted in and out. Test generation can proceed directly on the combinational part using any of the algorithms, such as PODEM, FAN and TOPS.

The sequential logic portion of the circuit is basically the scan-path flip-flops that must be tested first and separately from the combinational logic testing. Level Sensitive Scan Design, developed by IBM, is one of the most successful scan path techniques in use today.



Built-in Self -Test (BIST)

Built-In Self-Test is a design technique in which parts of a circuit are used to test the circuit itself. BIST is a cost-effective test methodology for highly complex VLSI chips like the Systems On Chip. BIST offers a superior solution to the test application problem. First, BIST circuitry can test chips, boards and the entire system without expensive external automatic test equipment. Second, we can use the same tests and test circuitry that we use for production testing whenever the device is subjected for testing. BIST schemes provide advantages such as at-speed testing and easy reuse of the built-in test structure for diagnosis and repair.

The basic 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 Read Only Memory (ROM) with stored patterns, a counter, Cellular Automata (CA) and a Linear Feedback Shift Register (LFSR). A typical response analyzer is a comparator with stored responses or an LFSR used as a signature analyzer. A test controller is necessary to activate the test and analyze the responses. Researchers proposed various types of BIST architectures viz., Centralized and Separate Board-Level (CSBL), Built-In Evaluation and Self-Test (BEST), Random-Test Socket (RTS), LSSD On-Chip Self Test (LOCST), Self-Test Using MISR and Parallel SRSG (STUMPS), Concurrent BIST (CBIST), Centralized and Embedded BIST architecture with Boundary Scan (CEBS), Random Test Data (RTD), Simultaneous Self-Test (SST), Cyclic Analysis Testing System (CATS), Circular Self-Test Path (CSTP). These architectures fall under centralized versus distributed BIST hardware and internal versus external BIST hardware classes.

Applying test patterns

There are many ways to generate and apply stimuli for the Circuit Under Test in a BIST scheme. Most widely used stimuli are the *exhaustive, pseudo-exhaustive* and *random* approaches.

An N-bit counter is a good example of an exhaustive pattern generator. However, as N increases, the testing time increases exponentially. Normally in a system with N input variables, every function may depend only on a subset of M inputs. Then the test pattern set which contains 2^M vectors gives the same degree of confidence in a system as if it is tested with all 2^N vectors since each function in the circuit is tested exhaustively. The testing time and power consumption during test can be reduced by shrinking the test pattern sequence. Another important technique for reducing the number of patterns is to partition the circuit into independent blocks that can be tested concurrently.

Using any of the test generation methods discussed previously, an effective set of "k" test vectors (patterns) can be identified. These "k" patterns can be applied to the CUT in the following ways.

- By choosing a suitable Linear Feedback Shift Register (LFSR) which can cycle through the minimum patterns in which all "k" patterns are included.
- By designing suitable Cellular Automata.

Autonomous circuits such as LFSRs are used as low-cost test pattern generators for circuits testable by pseudo-random patterns. LFSRs are widely used in BIST because they are simple and fairly regular in structure, their shift property integrates easily with serial scan, and they can generate exhaustive and/or pseudo random patterns. When these patterns are used as test sequences, it is possible to get different fault coverage in the circuit under test.

Programmable Cellular Automata (PCA) enables a single CA to be configured with different rules at different instants of time. A methodology using Binary Decision Diagram (BDD) to generate the set of all possible rules of CA and corresponding number of clock cycles for deterministic test sequence generation to a CUT is reported. Computation of PCA rule configuration requires solving a set of simultaneous boolean equations to generate the pattern P_{i+1} from P_i. To check whether the CA obtained by solving this set of equations can also realize the transition from P_{i+1} to P_{i+2}, these equations will be represented as BDDs. These BDDs are ANDed with the corresponding ones realizing transition from P_i to P_{i+1}. If any of these combined BDDs is unsatisfied, then we have to use another CA rule for making a pattern transition from P_{i+1} to P_{i+2}. Thus, the entire set of PCA rules required to generate the sequence is computed.

Response comparison/ signature analysis

The process of reducing the complete output response of a CUT into a signature is referred to as response compacting or compressing. Signature analysis is a data compression technique based on the concept of Cyclic Redundancy Checking (CRC). It can be realized as hardware using LFSRs. The signature is the contents of the register (i.e. LFSR) after the last input bit has been sampled by the LFSR. A circuit is tested by comparing the observed signature with the pre-computed (correct) signature.

Signature analysis has become popular when compared to other data com-

pression techniques viz., ones counting, transition counting, parity checking and syndrome checking. This is because it provides excellent fault and error coverage. Signature analysis can be extended to testing multiple outputs of a CUT through Multiple Input Signature Registers (MISR). In BIST schemes, signature analysis can be easily implemented using existing resources of the chip itself.

Conclusions

The efficiency of any method is evaluated in terms of fault coverage, test application time, power consumption, area overhead, performance penalty/degradation and computational time to arrive at a particular solution for testing a circuit. But testing also requires extremely expensive equipment. What's more, the time a chip spends in the tester results in a significant portion of the manufacturing cost of the chip.

Since test application time is proportional to the length of the test sequence that needs to be applied, it is desirable to apply a shorter test sequence that provides the desired fault coverage to reduce the test cost. To find effective test vectors, many algorithms and heuristics have been developed and used in ATPG tool development. In case of sequential circuits, DFT and BIST methods can be adopted for increasing testability of a circuit.

Area and test times are major overheads encountered when using BIST techniques. The goal set to a test engineer is to reduce the BIST area overhead without sacrificing the quality of the test. The increasing use of BIST is driven not only by the cost of the Automatic Test Equipment (ATE) problem. It also is driven by technical challenges of ATE to keep up with the rapidly increasing device speed, edge placement accuracy and stimulus-responses data rates. An ideal BIST scheme achieves not only very high fault coverage with minimum area overhead, low or zero performance degradation and minimum test time but also has low power consumption while testing.

Power dissipation during test application also has to be considered. Circuits are often designed to operate in two modes: *normal mode* and *test mode*. The activation of embedded test structures during test mode can make it possible for system registers to contain states that cannot be reached during normal operation. As a result, state transitions that are not possible during normal mode are often possible during test mode.

Thus, during testing, sequences that create much larger power dissipation are often applied to a circuit as compared to sequences that are applied during normal mode. As a result, excessive power dissipation during testing could prevent periodic testing of such a system. Power dissipation can be reduced by test vector ordering and switching off the power to the test area during normal operation of the circuit.

IEEE P1500 Standard for Embedded Core Test (SECT) is a recent development for testing the Intellectual Property (IP)-cores viz., central processing units, digital signal processing blocks, and communication modules of various kinds. In SoC, P1500, SECT facilitates easy integration and interoperability with respect to testing for manufacturing defects, especially when various cores of different developers are brought together into a one system chip.

However, new techniques are required, which can provide a systematic means for synthesizing low-overhead test architectures and compact test solutions for testing core-based SoCs. Finally, Signal Integrity (SI) problems viz., distributed delay variations, cross talk induced delay and logic errors, excessive voltage drop and swing on power nets, and substrate and thermal noise are becoming test challenges for today's deep sub micron process technologies.

Read more about it

• Srivaths Ravi et al., "Fault-Diagnosis-Based Technique for Establishing RTL and Gate-Level Correspondences", *IEEE Tran. On CAD of ICs and Systems,* pp.1414-1425, December 2001.

• Indradeep Ghosh et al., "Automatic Test Pattern Generation for Functional Register-Transfer Level Circuits Using Assignment Decision Diagrams," *IEEE Tran. On CAD of ICs and Systems*, pp. 402-415, March 2001.

• Prabir Dasgupta et al., "Cellular Automata Based Deterministic Test Sequence Generator for Sequential Circuits," *13th Intl. Conf. On VLSI Design*, pp. 544-548, January 2000.

• V.D.Agarwal, "A Tutorial on BIST," *IEEE Design & Test of Computers*, Vol.10,No.1, pp.73-82, March 1993.

• M.Abramovici, M.A. Breuer, and A.D.Friedman, *Digital Systems Testing and Testable Design*, Rockville, Computer Science, 1990.

About the authors

Prof. L.M. Patnaik is a Professor with the Indian Institute of Science, Bangalore, in the Department of Computer Science and Automation. His research interests have been in the areas of Parallel and Distributed Computing, Mobile Computing, Soft Computing, CAD of VLSI Circuits, and Real-Time Systems. In these areas, he has published over 330 papers in refereed International Journals and Conference Proceedings. He is Fellow of the IEEE, and The Third World Academy of Sciences. He has been awarded the IEEE Computer Society's 1999 Technical Achievement Award. He serves on the Editorial Boards of almost a dozen International Journals. He has served in various capacities for over 60 IEEE sponsored Conference Committees. He can be reached at lalit@micro.iisc.ernet.in.

Prof. H. S. Jamadagni is the Chairman of Centre for Electronic Design and Technology (CEDT) at the Indian Institute of Science, Bangalore. His area of work is Computer Networks and Embedded Systems. He is working for a number of industry sponsored Research and Development projects. He is involved in a national level education project. His email address is hsjam@cedt.iisc.ernet.in

Dr. V.K. Agrawal is working in ISRO since 1976. Currently, he is a Group Director, Control Systems Group, ISRO Satellite Centre. He has worked on the design and development of On-board computer systems for satellite with state-of-art technology. His research interests include Parallel processing, Fault tolerant computing, VLSI design and microprocessor based design.

B.K.S.V.L.Varaprasad received the B.E. degree in Computer Engineering from Andhra University, India in 1987 and MS degree from Indian Institute of Science in 2000. He is currently working towards his Ph.D. degree at Indian Institute of Science in mixed-signal testing area. Since 1988, he is working in ISRO Satellite Centre in the area of Testing of VLSI circuits, design of ASIC's for Spacecraft electronics and software development.