Chapter 1

Introduction

The reliability of electronic systems is no longer a concern to the military, aerospace and banking industries, where failure consequences may have catastrophic impact. Reliability and testing techniques have become of increasing interest to all other applications, such as computers, telecommunications, consumer products, and automotive industry, because of the following factors : (1) at present, electronic systems are becoming ubiquitous in the workplace and are now being used in harsher environments; (2) because of the proliferation of their use, users of electronic systems are not necessarily experienced people and may misuse the machines inadvertently; and (3) the continuous decrease in technology feature size, accompanied by an increase in system complexity and speed, has resulted in newer failure modes.

A key requirement for obtaining reliable electronic system is the ability to determine that the systems are error free. The majority of the hardware used today consists of digital circuits. A circuit must be tested to guarantee that it is working and continues to work according to specifications. Such testing detects failures due to the manufacturing defects. It can also detect many field failures due to aging, environmental changes, power supply fluctuations, and so on. Test pattern generation is a complex problem. Often, simulation patterns developed for design verification are augmented with patterns that are generated manually or by an automatic test pattern generator (ATPG) to obtain a complete test set, capable of detecting all faults in the circuit. This test also verifies the functionality of its logic.

The patterns are then applied to the circuit using automatic test equipment (ATE). Because of the complexity of testing processes, a design approach aimed at making digital circuits more easily testable has been formulated. This approach to design is known as *design for test* (DFT) or *design for testability*.

As a result of the continuous decrease in the minimum feature size of transistors, both device density and design complexity have steadily increased. The shift toward submicron technologies has allowed integrated-circuit(IC) designers to increase the complexity of their designs to the extent that an entire system can now be implemented on a chip. This new paradigm of *system on a chip* (SOC) has changed the approach to design and testing. To increase the design productivity, and hence to decrease time-to-market, the reuse of previously designed modules is becoming common practice in SOC design. This is known as *core-based-design.* The reuse approach is not limited to in-house design, but is extended to modules that have been designed by others. Such modules are referred to as *embedded cores.*

An integrated circuit core is a predesigned, preverified silicon circuit block, usually containing at least few thousand (5000) gates, that can be used in building a large or more complex application on a semiconductor chip. A core may be *hard, firm* or *soft.*

Hard cores are optimized for area and performance and they are mapped into specific technology and possibly a specific foundry. They are ready to be dropped into a system. They are provided as layout files that can't be modified by the user. *Firm cores* are usually provided as technology-dependent netlists using library cells whose size, aspect ratio, and pin location can be changed to meet the customer's needs. Soft core consists of a synthesizable *hardware description language* (HDL) description that can be retargeted to different semiconductor processes.

Major problem in the core based system design is the adoption of efficient test and diagnostic strategies. Testing core-based System-on-Chips poses two-fold challenge [13], namely core-level testing and chip-level testing. Core-level testing involves making each core testable inserting the necessary *Design for Testability* (DFT) structures and generating test sequences. When the cores are integrated into a SOC, chip-level testing needs to be addressed by the SOC designer. The main difficulty in chip-level testing is the problem of justifying precomputed test sequences of a core embedded deep in the design from the

chip inputs, and propagating the test responses from the core outputs to the chip outputs. As the cores are deeply embedded in SOC, special access mechanisms are required to test them at system level, known as *Test Access Mechanisms* (TAMs). The efficiency of a TAM depends on to what extent it can reduce the testing time, which is the time to test all cores and interconnects between the cores in a SOC.

The cost of Automated Test Equipment (ATE) grows significantly with the increase in operating frequency, channel capacity and memory requirements of the test data [1]. Reduction in test data volume not only reduces ATE memory requirements but also lowers the testing time as much less amount of data would need to be transferred from ATE to the chip. There exist two potential solutions to alleviate the raising costs of ATE. The first solution is Built-In-Self-Test (BIST) [19] incorporated into the System-on-Chip. However, the application of BIST is limited; first due to pseudo-random resistant faults which limit the fault coverage achievable by BIST and secondly and most importantly due to the fact that currently there are very few cores that include BIST features. To incorporate BIST into the existing cores would require a considerable redesign effort. The second solution is that of compressing the test data stored on the tester (ATE) and utilizing small amount of *on-chip* hardware to decompress the data before being fed into the scan chains driving the *circuit-under-test* (CUT). This method not only reduces the memory requirements on the ATE but also reduces the testing time since the reduced volume of test data can be transferred faster to the chip. Moreover, unlike BIST, this method does not require the redesign of intellectual property (IP) cores. Test Data Compression requires compression techniques that are able to simultaneously satisfy the dual objectives of *loss-less compression* of test data with *minimal decompression architecture* circuit area overhead.

One of the other most important challenges in the field of SOC testing is of increased power dissipation during test. During testing, power consumption is much higher than during normal operational mode [37]. Increased power

consumption results in higher heat dissipation which has the potential to damage the circuit-under-test thereby decreasing the manufacturing yield. The situation is further worsened by the fact that several test scheduling techniques attempt to reduce testing time by testing several cores simultaneously [9].

Contribution of the thesis

In the above background, we have addressed some of the important problems related to SOC testing. These are as follows.

1. Designing a test access mechanism by partitioning the test lines available and schedule the cores for testing.
2. An integrated approach to reduce the overall testing time taking into consideration the core and interconnect testing. We have also made a trade-off between the power consumption and test time.
3. Test data compression to have improved compression ratio, reduced test application time with some extra added hardware.

Organization of the thesis

The thesis is organized as follows.

Chapter 2 presents a detailed survey of the works carried out in the field of SOC testing.

Chapter 3 presents a simulated annealing based approach for the partitioning of the test lines and assignment of cores to one of the partitions ensuring reduced test time. It shows improved results compared to many of the contemporary works.

Chapter 4 presents a genetic algorithm based approach to solve the combined problem of minimizing the test time and test power in SOC testing. The test time minimization not only addresses the core test time reduction, but also takes care of the reduction in the time required to test the interconnects.

Chapter 5 presents another core test scheduling solution that does not partition the test lines. It rather views the problem as a two-dimensional rectangle-packing one. It also takes into consideration the power constraint, in the sense that the overall power required at any point of time during testing should not exceed the specified power limit. The precedence constraints between the cores to be tested have also been addressed.

Chapter 6 presents a test data compression strategy that decomposes an input file into two to achieve higher compression ratio and reduced test application time, though at the cost of slightly higher area overhead.

Finally, Chapter 7 draws the conclusion and enumerates the scopes for future works.