WAFER-LEVEL TESTING AND TEST PLANNING FOR INTEGRATED CIRCUITS

by

Sudarshan Bahukudumbi

Department of Electrical and Computer Engineering
Duke University

Date: __________________________
Approved:

____
Prof. Krishnendu Chakrabarty, Chair

____
Prof. John Board

____
Prof. Romit Roy Choudhury

____
Prof. Montek Singh

____
Prof. Kishor Trivedi

Dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the Graduate School of
Duke University

2008

ABSTRACT

WAFER-LEVEL TESTING AND TEST PLANNING FOR INTEGRATED CIRCUITS

by

Sudarshan Bahukudumbi

Department of Electrical and Computer Engineering
Duke University

Date: ______________________
Approved:

____
Prof. Krishnendu Chakrabarty, Chair

____
Prof. John Board

____
Prof. Romit Roy Choudhury

____
Prof. Montek Singh

____
Prof. Kishor Trivedi

An abstract of a dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the Graduate School of
Duke University

2008

Copyright © 2008 by Sudarshan Bahukudumbi
All rights reserved

Abstract

The relentless scaling of semiconductor devices and high integration levels have lead to a steady increase in the cost of manufacturing test for integrated circuits (ICs). The higher test cost leads to an increase in the product cost of ICs. Product cost is a major driver in the consumer electronics market, which is characterized by low profit margins and the use of a variety of core-based system-on-chip (SoC) designs. Packaging has also been recognized as a significant contributor to the product cost for SoCs. Packaging cost and the test cost for packaged chips can be reduced significantly by the use of effective test methods at the wafer level, also referred to as wafer sort.

Test application time is a major practical constraint for wafer sort, even more than for package test. Therefore, not all the scan-based digital test patterns can be applied to the die under test. This thesis first presents a test-length selection technique for wafer-level testing of core-based SoCs. This optimization technique, which is based on a combination of statistical yield modeling and integer linear programming (ILP), provides the pattern count for each embedded core during wafer sort such that the probability of screening defective dies is maximized for a given upper limit on the SoC test time. A large number of wafer-probe contacts can potentially lead to higher yield loss during wafer sort. An optimization framework is therefore presented to address test access mechanism (TAM) optimization and test-length selection for wafer-level testing, when constraints are placed on the number of number of chip pins that can be contacted.

Next, a correlation-based signature analysis technique is presented for mixed-signal test at the wafer-level using low-cost digital testers. The proposed method overcomes the limitations of measurement inaccuracies at the wafer-level. A generic cost model is developed to evaluate the effectiveness of wafer-level testing of analog and digital cores in a mixed-signal SoC, and to study its impact on test escapes, yield loss and packaging cost. Results are presented for a typical mixed-signal "big-D/small-A" SoC from industry, which contains a large section of flattened digital logic and several large mixed-signal cores.

Wafer-level test during burn-in (WLTBI) is an emerging practice in the semiconductor industry that allows testing to be performed simultaneously with burn-in at the wafer-level. However, the testing of multiple cores of a SoC in parallel during WLTBI leads to constantly-varying device power during the duration of the test. This power variation adversely affects predictions of temperature and the time required for burn-in. A test-scheduling technique is presented for WLTBI of core-based SoCs, where the primary objective is to minimize the variation in power consumption during test. A secondary objective is to minimize the test application time.

Finally, this thesis presents a test-pattern ordering technique for WLTBI. The objective here is to minimize the variation in power consumption during test application. The test-pattern ordering problem for WLTBI is solved using ILP and efficient heuristic techniques. The thesis also demonstrates how test-pattern manipulation and pattern-ordering can be combined for WLTBI. Test-pattern manipulation is carried out by carefully filling the don't-care (X) bits in test cubes. The X-fill problem is formulated and solved using an efficient polynomial-time algorithm.

In summary, this research is targeted at cost-efficient wafer-level test and burn-in of current- and next-generation semiconductor devices. The proposed techniques are expected to bridge the gap between wafer sort and package test, by providing cost-effective wafer-scale test solutions. The results of this research will lead to higher shipped-product quality, lower product cost, and pave the way for known good die (KGD) devices, especially for emerging technologies such as three-dimensional integrated circuits.

Acknowledgements

There are several people who significantly influenced this dissertation - in ways direct and indirect - and I would like to thank them here. My advisor, Dr. Krishnendu Chakrabarty provided me the academic freedom to pursue research problems that truly interested me, and for that I am very grateful. His genuine interest in my progress, technical insights and pursuit of perfection have largely been responsible for making me a better researcher.

I thank Dr. Sule Ozev for providing valuable counsel and feedback on the mixed-signal project, and for educating me on the practical aspects of mixed-signal testing. I would also like to thank Vikram Iyengar of IBM Corporation for providing industry-insights on the mixed-signal project and for his help in preparing the mixed-signal manuscript. I thank Rick Kacprowicz of Intel Corporation for being our industrial mentor in the burn-in project, and for providing valuable insights on the implementation aspects of our work.

I would like to thank my committee members Dr. Kishor Trivedi, Dr. John Board, Dr. Montek Singh, and Dr. Romit Roy Choudhury for taking time to serve on my dissertation committee, and for providing constructive technical feedback on my work. I would also like to thank Dr. Chris Dwyer for serving on my preliminary examination committee.

I would like to thank people in my research group Zhanglei Wang, Lara Oliver, Mahmut Yilaz and Yang Zhao. I have benefited greatly from numerous discussions with Zhanglei and Mahmut on a wide range of topics- from testing to politics. I am also indebted to Mahmut and Hongxia Fang for all their help with data generation for my research projects.

Many people in the secretarial and support staff in electrical engineering specifi-

cally Autumn Wenner and Ellen Currin have helped me on numerous occasions with travel reimbursements, departmental letters and administrative support, making my life around here a lot easier.

I am grateful for financial support I received for my graduate studies from the Semiconductor Research Corporation and the National Science Foundation.

Finally, I would like to thank my mom, dad and brother for being a constant source of support and comfort in times of need. This dissertation will not be complete without the excellent support system they have provided over the years.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

As predicted by Moore's law, the number of transistors on a chip has continued to grow in recent years. This increase in transistor count has been accompanied by rapid advances in the semiconductor industry, such as higher levels of integration on a chip, greater functionality, faster clock rates, lower device power, and small form factors. Shrinking feature sizes drive down the cost per transistor, a trend that has been reported recently in the International Technology Roadmap for Semiconductors (ITRS) [1]. The cost of manufacturing test, however, has failed to follow this trend; the cost of test per transistor has shown no appreciable decrease over time. This trend can clearly be seen from Figure 1.1 [1], where the cost for manufacturing and testing a transistor are illustrated on the same axes. Higher levels of integration on a chip lead to significant increase in test time. The increase in test time for these devices leads to an increase in the test cost. More research is therefore needed to reduce the test cost per transistor of integrated circuits (ICs).

A system-on-chip (SoC) integrated circuit consists of a set of complex pre-designed modules, referred to as embedded cores, which are implemented on the same piece of silicon [4]. An SoC provides the system integrator with a wider variety of design alternatives than earlier generations of application-specific ICs. Recent advances in semiconductor process technologies and the advent of sophisticated design tools have enabled the design of complete electronic systems on a single chip. In order to handle complexity and satisfy the ever-increasing demand for shorter time-to-market for SoCs, design engineers typically use pre-designed and pre-verified embedded cores in their designs. These embedded cores, typically provided by "fabless" companies

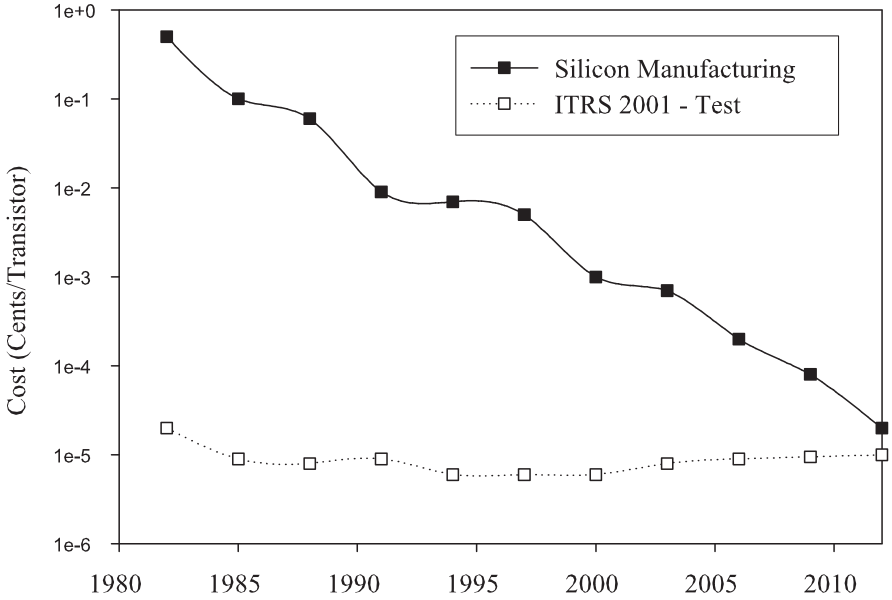


Figure 1.1: Trend in test cost versus manufacturing cost per transistor (adapted from [1]).

in black-box form, are known as intellectual property (IP) cores.

The manufacturing test of SoCs is a process where test stimuli are applied to the fabricated SoC by means of a test-access mechanism (TAM). The TAM provides test access to the embedded cores in the SoC from the input/output (I/O) terminals of the chip. The steps involved in testing of SoCs, and semiconductor devices in general, can be classified into three categories: wafer sort or probe test, post-package manufacturing test, and burn-in.

Wafer sort is the first step in the manufacturing test process, where the chip in bare wafer form is tested for manufacturing defects. The devices are subjected to standardized parametric and functional tests; devices that pass these tests are subjected to further assembly and test processes, and the ones that fail these tests are marked with an ink dot on the wafer to indicate that they are faulty.

Once the devices that pass the test at the wafer-level are packaged, they are

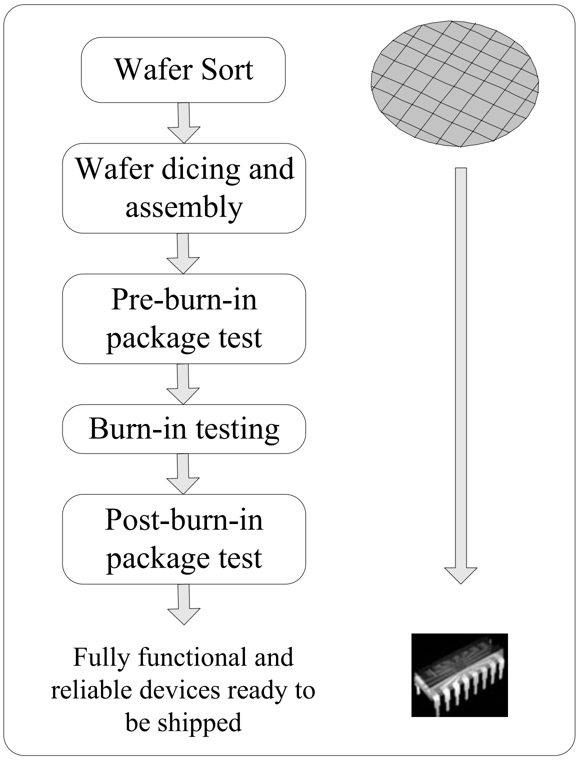


Figure 1.2: The steps involved in the testing of a semiconductor device.

subjected to package test. The package test process is often carried out in two stages. The first stage of testing a packaged device takes place before the burn-in process, and the second stage, i.e., the final step in testing the device, is carried out after burn-in. Complete parametric, functional, and structural testing are performed during package testing of these devices.

Some devices that pass all the manufacturing tests may fail early during their lifetime. The burn-in test process accelerates failure mechanisms that cause early field failures ("infant mortality") by stressing packaged chips under high operating temperatures and voltages. The burn-in process is therefore an important component in the test and manufacturing flow of a semiconductor device, and it is necessary to ensure reliable field operation. Figure 1.2 illustrates the conventional test flow for semiconductor devices.

Techniques and solutions employed for probe testing of SoCs can also be used exclusively during the manufacture of known good dies (KGDs); KGDs are fully functional devices that are sold as bare dies and used in the manufacture of complex system-in-package (SiP) devices and multi-chip packages (MCPs). Until recently, a major concern was the electrical integrity of the bare die. There are several challenges to performing full electrical testing of a bare die to verify conformance to specifications. Also, until recently, bare die were not subjected to burn-in. Thus latent defects went undetected with the bare die. With recent advances in the manufacture of semiconductor test equipment, and increased awareness of the importance of the KGD, complex test and burn-in functions can be carried out at the wafer level [5, 6].

Market and functionality segments exist for both SoCs and KGD integration in SiPs and three-dimensional (3-D) ICs, and these design approaches are complementary rather than competitive [3]. SoCs find applications in standardized processes for digital-centric functions, thereby enabling easy and seamless integration of additional functions when necessary. SiPs and stacked 3-D ICs provide an approach where a mix of devices, components, and technologies are used to maximize performance and cost. Designers are thus able to drastically reduce the time-to-market with the choice of such design technologies. It is therefore important to address the test challenges of SoCs as well as reduce the test cost for the manufacture of KGDs at the wafer level.

1.1 Background

In this section, we review some key testing methods and concepts that are referred to in the rest of the thesis.

1.1.1 System-level design-for-test and test scheduling for core-based SoCs

The testing of core-based SoCs requires the availability of a suitable on-chip test infrastructure, which typically include test wrappers and TAMs. A core test wrapper is the logic circuitry that is added around the embedded core to provide suitable test access to the core, while at the same time isolating the core from its surrounding logic during test [7, 8, 9]. The test wrapper provides each core with a normal mode of operation, an external-test mode, and an internal-test mode. When the core is in the normal mode of operation, it maintains the functionality that is desired for proper device operation; the wrapper is transparent to the surrounding logic in this mode of operation. The core in its external-test mode observes the wrapper elements for interconnect test, and when the core is in the internal-test mode, the wrapper elements control the state of the core input terminals for testing the core internal logic. The TAM transports test stimuli and responses between the SoC pins and the core terminals. Careful design of test wrappers and TAM can lead to significant cost savings by minimizing the overall test time [8, 9, 10, 11, 12, 13, 14].

Figure 1.3 illustrates the use of generic core test wrappers and TAMs for a design with N embedded cores [2]. The test source provides test vectors to the embedded cores via on-chip linear feedback shift registers (LFSRs), a counter, ROM, or off-chip automatic test equipment. A test sink, by means of on-chip signature analyzers, or off-chip automatic test equipment (ATE), provides verification of the output responses. The TAM is user-defined; the system integrator must design these structures for the SoC by optimally allocating TAM wires to the embedded cores in the SoC with the objective of minimizing the overall test time. The TAM is not only used to transport test stimuli and responses to and from the cores, but it is also used for interconnect test between the embedded cores in the SoC. The test access port (TAP) receives

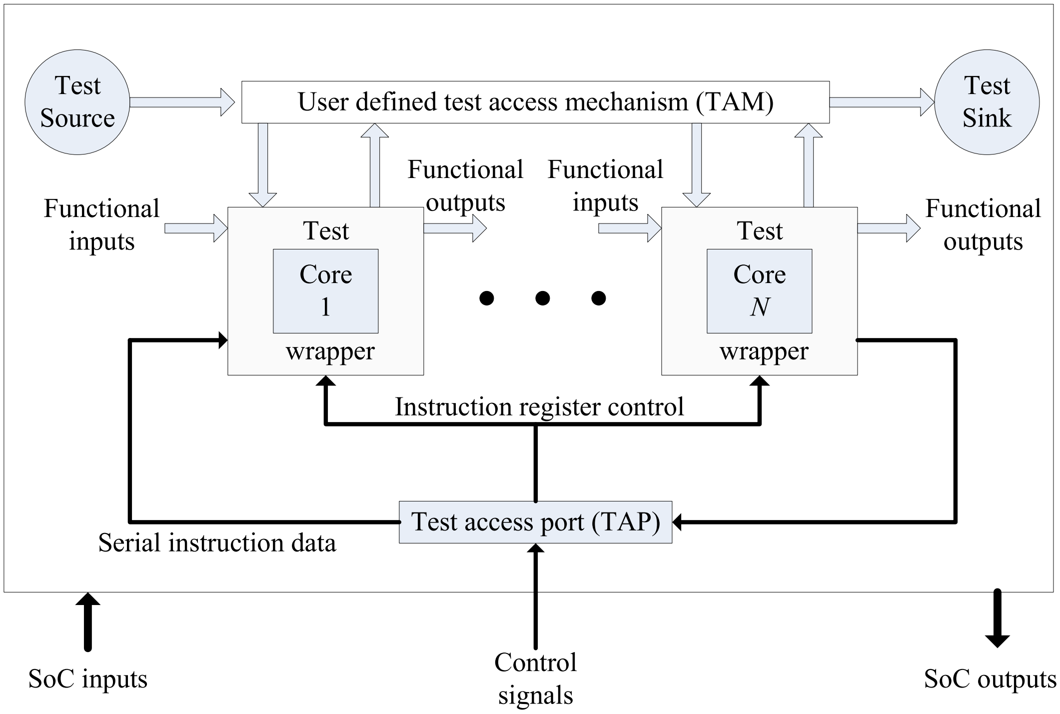


Figure 1.3: Test architecture based on wrappers and a TAM [2].

control signals from outside to control the mode of operation of test wrappers; the TAP enables the loading of test instructions serially on to the test-wrappers.

The testing of core-based SoCs continues to be a major concern in the semiconductor industry [1, 15]. The recent IEEE 1500 Standard addresses some aspects of the testing of core-based SoCs [7]. A standardized 1500 wrapper can either be provided by the core vendor or it can be implemented during system integration. Wrapper/TAM co-optimization in conjunction with test scheduling play an important role during system integration as they directly impact the testing time for the SoC and the associated tester data volume. There are several issues to be considered during test scheduling of core-based SoCs, e.g., power consumption constraints [16, 17], precedence constraints during test [17, 18], conflicts between cores arising from the use of shared TAM wires, etc. A number of efficient solutions have recently been proposed for TAM optimization and test scheduling [8, 11, 12, 14, 16, 18, 19];

however, these methods are aimed at reducing the test time for package test only. They do not address the problems that are specific to wafer-level testing.

1.1.2 Wafer-level test during burn-in

In addition to the need for effective test techniques for defect screening and speed binning for ICs, there is an ever-increasing demand for high device reliability and low defective-parts-per-million levels. Semiconductor manufacturers routinely perform reliability screening on all devices before shipping them to customers [20]. Accelerated test techniques shorten time-to-failure for defective parts without altering the device failure characteristics [21]. Burn-in is one such technique that is widely used in the semiconductor industry [6, 21].

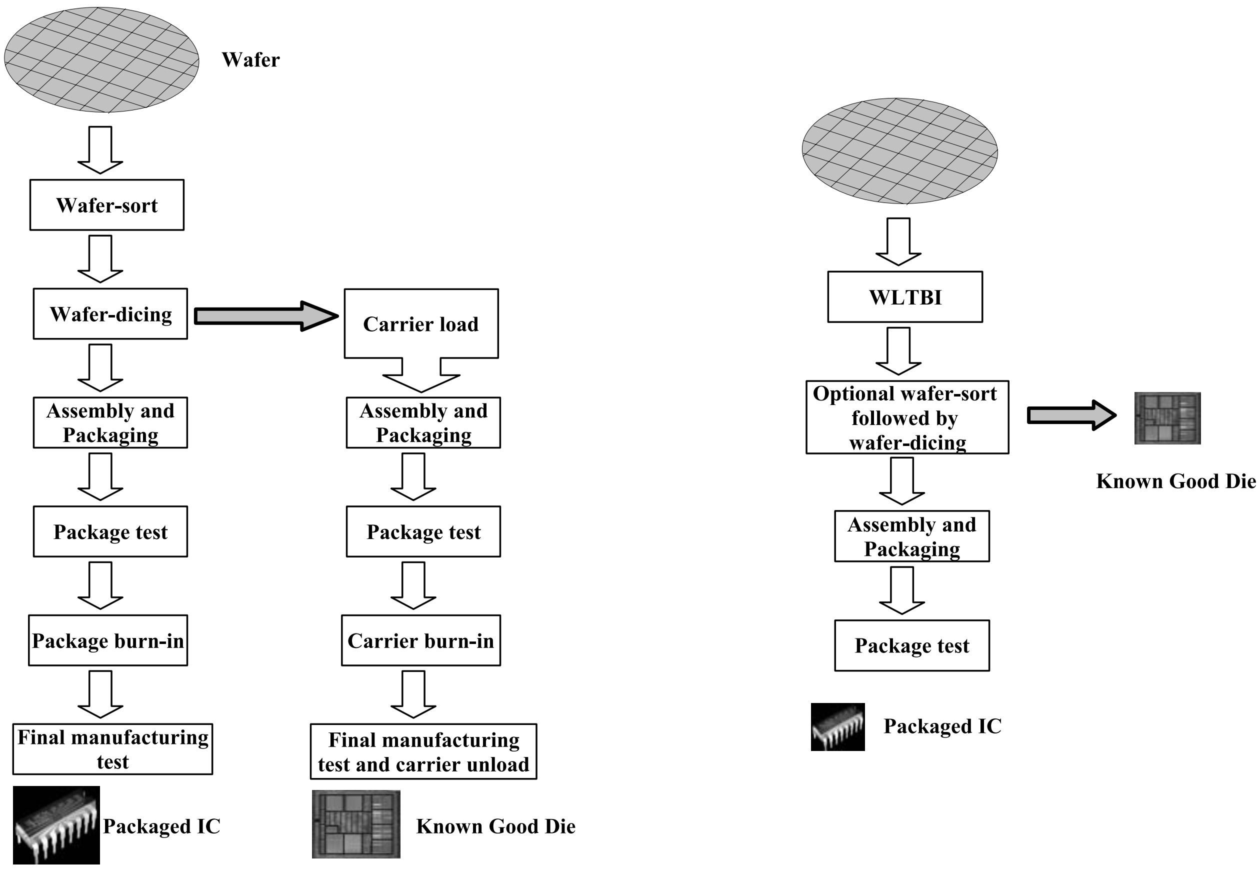
The long time intervals associated with burn-in result in high cost [1, 22, 23]. It is however unlikely that burn-in will be completely eliminated in the near future for high-performance chips and microprocessors [1]. Wafer level burn-in (WLBI) has recently emerged as an enabling technology to lower the cost of burn-in [6]. In this approach, devices are subjected to burn-in and electrical testing while in the bare wafer form. By moving the burn-in process to the wafer-level, significant cost savings can be achieved in the form of lower packaging costs, as well as reduced burn-in and test time.

Test during burn-in at the wafer-level enhances the benefits that are derived from the burn-in process. The monitoring of device responses while applying suitable test stimuli during WLBI leads to the easier identification of faulty devices. We refer to this process as “wafer-level test-during-burn-in” (WLTBI); it is also referred to in the literature as “test in burn-in” (TIBI) [21], “wafer-level burn-in test” (WLBT) [24], etc.

Figure 1.4 illustrates and compares the test and burn-in flow in a semiconductor manufacturing process. The manufacturing flow for package-level burn-in (PLBI) is shown in Figure 1.4(a); Figure 1.4(b) highlights the manufacturing flow when WLTBI is employed for test and burn-in at the wafer-level. Test and burn-in of devices in the bare wafer form can potentially reduce the need for post-packaging test and burn-in for packaged chips and KGDs. In the manufacture of KGDs, WLTBI eliminates the need for a die-carrier and carrier burn-in, thereby resulting in significant cost savings.

The basic techniques used for the testing and burn-in of individual chips are the same as those used in WLTBI. Test and burn-in require the availability of suitable electrical excitation of the device/die under test (DUT), irrespective of whether it is done on a packaged chip or a bare die. The only difference lies in the mode of delivery of the electrical excitation. Mechanically contacting the leads provides electrical bias and excitation during conventional testing and burn-in. In the case of WLTBI, this excitation can be provided in any of the following three ways: the probe-per-pad method, the sacrificial metal method and the built-in test/burn-in method [25].

The built-in test/burn-in method involves the use of on-chip design-for-test (DfT) infrastructure to achieve WLTBI. This technique allows wafers to undergo full-wafer contact using far fewer probe contacts. The presence of sophisticated built-in DfT features on modern day ICs makes "monitored burn-in" possible. *Monitored burn-in* is a process where a DUT is provided with input test patterns; the output responses of the DUT are monitored on-line, thereby leading to the identification of failing devices. It is therefore clear that WLTBI has a significant potential to lower the overall product cost by breaking the barrier between burn-in and test processes. As a result, ATE manufacturers have recently introduced WLBI and test equipment that provide full-wafer contact during burn-in and they also provide test monitoring capabilities [6, 24, 26].



(a) (b)

Figure 1.4: Test and burn-in flow using: (a) PLBI; (b) WLTBI.

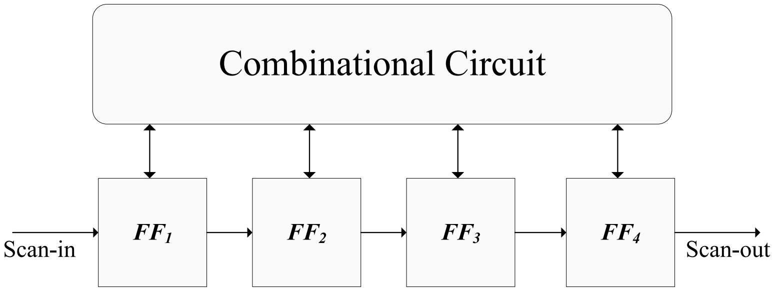


Figure 1.5: Flip-flops in a circuit connected as a scan chain.

1.1.3 Scan design

Scan design is a widely used DfT technique that provides controllability and observability for flip-flops by adding a scan mode to the circuit. When the circuit is in scan mode, all the flip-flops form one or more shift registers, also known as scan chains. Using separate scan access I/O pins, test patterns are serially shifted in to the scan chains and test responses are serially shifted out. This process significantly reduces the cost of test by transforming the sequential circuit into a combinational circuit for test purposes. For circuits with scan designs, the test process involves test pattern application from external ATE to the primary inputs and scan chains of the DUT. To make a pass/fail decision on the DUT, the states of the primary outputs and the flip-flops are fed back to the ATE for analysis. Figure 1.5 illustrates how flip-flops are connected to form a scan chain.

1.2 Motivation for thesis research

The ATE is first used in the semiconductor manufacturing process during wafer sort, when the chip is still in the bare wafer form. Effective defect screening at the wafer level leads to significant cost savings by eliminating the assembly and further testing of faulty die. Data generated during the sort process quickly provides valuable feedback to the wafer fab. This information is time-sensitive, and the timely reporting of this information to the fab can facilitate changes to the manufacturing process that can increase the yield.

1.2.1 Challenges associated with wafer sort

Wafer-level testing leads to early defect screening, thereby reducing packaging and production cost [2, 27, 28]. As highlighted in [1, 29], packaging cost accounts for a significant part of the overall production cost. Current packaging costs for a cost-sensitive, yet performance-driven, IC can vary between \$3.6 to \$20.5, depending on the number of pins in the IC [1]. These costs are further increased for high-performance ICs. It has also been reported that the packaging cost per pin exceeds the cost of silicon per square millimeter, and the number of pins per die can easily exceed the number of square millimeters per die [1, 29].

Several challenges are associated with testing at the wafer-level. These challenges need to be addressed in order to reduce the cost associated with the complete test process of a semiconductor chip.

Semiconductor companies often resort to the use of low-cost testers at the wafer-level to reduce the overall capital investment on ATE. These testers are constrained by the limited amount of memory available to store test patterns and responses, the number of available tester channels, and the maximum frequency at which they can be operated. Reduced memory and the limited the number of available tester channels reduce the number of devices that can be tested simultaneously. This is especially a severe limitation at the wafer-level since there are multiple dies on a single wafer; decrease in parallelism due to tester limitations results in a significant increase in the overall test time.

Measurement inaccuracies are common when analog cores are tested in a mixed-signal test environment based on digital signal processing. This problem is exacerbated by noisy DC power supply lines, improper grounding of the wafer probe, and lack of proper noise shielding of the wafer probe station [30]. The above problems make test and characterization at the wafer-level especially difficult, and they can lead to high yield loss during wafer sort.

The scaling of test costs for semiconductor devices highlights the need for new techniques to minimize the overall test cost. Several techniques to minimize the overall test time for SoCs during package testing have been proposed in [8, 10, 11, 12, 9]. In contrast, test planning for effective utilization of hardware resources for wafer-level has not been studied. There is a need for basic research in two focus areas related to wafer-level testing of core-based digital SoCs. It is common practice in industry to partially test these devices at the wafer level in order to reduce test cost. The first focus area addresses wafer-level test planning of these devices under constraints of test application time. The ATE is also constrained by the number of available tester channels because of the use of low-cost digital testers. The second focus area develops test techniques to test these devices at the wafer-level under such limitations.

In a special class of SoC designs known as "big-D/small-A" mixed-signal SoCs, the fraction of die area taken up by analog circuits can range from 5% to 30% [31]. The DragonBall^{TM}-MX1 SoC, details for which are presented in [32], is an example of a "big-D/small-A" mixed-signal SoC. Most "big-D/small-A" SoCs comprise of at least a pair of complementary data converters, a significant amount of digital logic and a PLL [32, 33]. In the SoC described in [32], the mixed-signal components constitute up to 10% of the overall die area. The applications of mixed-signal SoCs to the consumer market are numerous, ranging from medical monitoring devices to audio products and handheld devices. The consumer electronics market is also characterized by low profit margins and rising packaging costs [1, 29]. Test and packaging costs are therefore of increasing importance for such SoCs. Wafer-level defect screening techniques to test these devices are essential in order to minimize the overall test cost.

1.2.2 Emergence of KGDs

Wafer sort testing was once considered a method to save packaging costs by eliminating bad dies. Today, wafer sort is an important step in process control, yield enhancement, and yield management [34]. The emerging trend of selling bare dies (KGDs) instead of packaged parts further emphasizes the importance of wafer sort. KGDs are handled in the following ways: a) packaged by the customer in a custom package; b) mounted directly on a substrate; c) combined with other die in a MCP or SiPs [34]. KGDs produced with different process technologies can be integrated into a high-density product at the package level.

With the emergence of MCPs and SiPs, the yields of the individual die making up the package determine the overall yield of the product. Full functional and structural testing of these devices at wafer sort is therefore important. In addition to testing these devices, there is a need to burn-in these devices in their bare wafer form to weed out all latent defects and ensure reliable operation.

The test flow for a typical SiP, shown in Figure 1.6, highlights the need for cost-effective wafer-scale test and burn-in solutions.

1.2.3 WLTBI: Industry adoption and challenges

WLTBI technology has recently made rapid advances with the advent of the KGD [35]. The growing demand for KGDs in complex SoC/SiP architectures, multi-chip modules, and stacked memories, highlights the importance of cost-effective and viable WLTBI solutions [6]. WLTBI will also facilitate advances in the manufacture of 3-D ICs, where bare dies or wafers must be tested before they are vertically stacked.

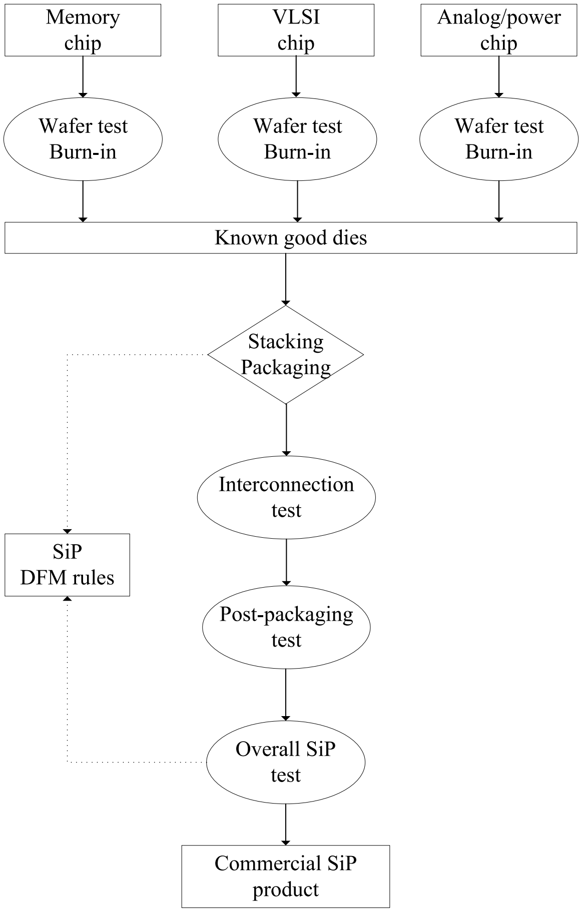


Figure 1.6: System-in-package test flow [3].

WLTBI can therefore be viewed as an enabling technology for cost-efficient manufacture of reliable 3-D ICs.

Recently, Motorola teamed up with Tokyo Electron Ltd. and W.L. Gore & Associates Inc. to develop a WLTBI system for commercial use. These systems provided a full-wafer direct-contact solution for bumped die used in flip-chip assembly applications [25]. Aehr Test Systems recently announced that it shipped full wafer contact burn-in and test systems to a leading automotive IC manufacturer [36]. These test systems have the ability to contact 14 wafers simultaneously by providing 30,000 contact-point capability per wafer. The test features of the system include a full algorithmic test for memories, and vector pattern generator for devices using BIST [36].

A study was presented in [5] to compare the cost of wafer level burn-in and test with the burn-in and test of a singulated die. It was shown that in a high-volume manufacturing environment, wafer-level burn-in and test was more cost-effective compared to equivalent “die-pack” and chip-scale packaging technologies [5]. Similar WLBI and TDBI equipment with response monitoring capabilities at elevated temperatures have been successfully manufactured and deployed by other leading test equipment manufacturing firms such as Advantest [37] and Delta-V instruments [6].

Successful implementation of WLTBI technologies is essential for the cost-effective manufacture of devices used in communication, automobile and computer markets [25]. The following are some of the benefits of WLTBI, which motivate the need for enabling technologies for WLTBI.

1. Burn-in for KGDs at the wafer level require less test insertions and also reduces the burn-in cycle time when compared with die-level burn-in [25].
2. WLTBI can provide quick feedback to wafer manufacturing; this provides an effective mechanism for process control, while at the same time improving the yield of the process.
3. It has been shown in [5] that WLTBI is a cost-efficient technique for the manufacture of reliable and fully functional KGDs.
4. Commercial WLTBI test equipment are currently being deployed by some leading semiconductor companies to lower manufacturing cost [5].

Thermal Challenges

Successful WLTBI operation requires a through understanding of the thermal characteristics of the DUT. In order to keep the burn-in time to a minimum, it is essential to test the devices at the upper end of their temperature envelope [38]. Moreover, the junction temperatures of the DUT need to be maintained within a small window such that burn-in predictions are accurate.

Scan-based testing is now widely used in the semiconductor industry [39]. However, scan testing leads to complex power profiles during test application; in particular, there is a significant variation in the power consumption of a device under test on a cycle-by cycle basis. In a burn-in environment, the high variance in scan power adversely affects predictions on burn-in time, resulting in a device being subjected to excessive or insufficient burn-in. Incorrect predictions may also result in thermal runaway.

The challenges that are encountered during WLTBI are a combination of the problems faced during the sort process and during burn-in. Wafer-sort is used to identify defective dies at wafer level before they are assembled in a package. It is also the first test-related step in the manufacturing process where thermal management plays an important role. Current wafer probers use a thermal chuck to control the device temperature during the sort process. The chuck is an actively regulated metal device controlled by external chillers and heaters embedded under the device [38]. The junction temperature of the DUT is determined by the following relationship [38, 40, 41]:

$$T_j = T_a + P \cdot \theta_{ja} \tag{1.1}$$

where T_j is the junction temperature of the device, T_a is the ambient temperature, P is the device power consumption, and θ_{ja} is the thermal resistance (junction to ambient) of the device. The value of T_j is therefore determined by the device power consumption, thermal resistance, and a constant T_a. The controllability of T_j is limited by the extent to which the parameters T_a and P can be controlled. Considerable power fluctuations during the test of the DUT can significantly affect the value of T_j for the DUT, thereby adversely impacting the reliability screening process.

One of the important goals of the burn-in process is to keep the burn-in time to a minimum, thereby increasing throughput, and minimizing equipment and processing costs. It is also important to have a tight spread in temperature distribution of the device, to increase yield and at the same time minimize burn-in time [38]. The parameter T_j cannot exceed a pre-determined threshold due to concerns of thermal runaway and the need to maintain proper circuit functionality. It is this issue of controlling the spread in T_j over the period of test application that we address in this thesis.

The objective of this thesis research is to reduce the overall cost of the product by efficient test planning and test resource optimization at the wafer level. Four key research problems are identified and solved in this thesis.

- **Wafer-level test planning for core-based digital SoCs.** A framework for TAM optimization and test-length selection for wafer-level testing of core-based digital SoCs is necessary, especially when constraints are placed on the number of chip pins that can be contacted and the overall test time for the SoC during probe test.

- **Wafer-level defect-screening for mixed-signal SoCs.** The goal here is to develop a signature analysis technique that is especially suitable for mixed-signal test at the wafer-level using low-cost digital testers.

- **Test scheduling for WLTBI of core-based SoCs.** An efficient test-scheduling method is needed for core-based SoCs that reduces the overall variation in power consumption during WLTBI.

- **Test-pattern ordering and test-data manipulation for WLTBI.** The goal here is to optimally order test patterns and carefully fill the don't care bits in the test cube such that the variation in power consumption during WLTBI is minimized.

We present additional background material on the above problem areas in Sections 1.3-1.6, and also provide a brief overview of the work carried out in this thesis.

1.3 Wafer-level test planning for core-based SoCs

A recent SoC test scheduling method attempted to minimize the average test time for a packaged SoC, assuming an abort-on-first fail strategy [42, 43]. The key idea in this work is to use defect probabilities for the embedded cores to guide the test scheduling procedure. These defect probabilities are used to determine the order in which the embedded cores in the SoC are tested, as well as to identify the subsets of cores that are tested concurrently. The defect probabilities for the cores were assumed in [42] to be either known *a priori* or obtained by binning the failure information for each individual core over the product cycle [43]. In practice, however, short product cycles make defect estimation based on failure binning difficult. Moreover, defect probabilities for a given technology node are not necessarily the same for the next (smaller) technology node. Therefore, a yield modeling technique is needed to accurately estimate these defect probabilities.

Test time is a major practical constraint for wafer sort, even more so than for package test, because not all the scan-based digital tests can be applied to the die under test. It is therefore important to determine the number of test patterns for each core that must be used for the given upper limit on SoC test time for wafer sort, such that the probability of successfully screening a defective die is maximized. The number of patterns need to be determined on the basis of a yield model that can estimate the defect probabilities of the embedded cores, as well as a "test escape model" that provides information on how the fault coverage for a core depends on its test length.

Reduced-pin count testing (RPCT) has been advocated as a design-for test technique, especially for use at wafer sort, to reduce the number of IC pins that needs to be contacted by the tester [44, 45, 46, 47, 48]. RPCT reduces the cost of test by enabling the reuse of old testers with limited channel availability. It also reduces the number of probe points required during wafer test; this translates to lower test cost, as well as less yield loss issues arising from contact problems with the wafer probe. We have developed an optimization framework for wafer sort that addresses TAM optimization and test-length selection for wafer-level testing of core-based digital SoCs.

1.4 Wafer-level defect screening for mixed-signal SoCs

The test cost for a mixed-signal SoC is significantly higher than that for a digital SoC [49]. This is due to the capital cost associated with expensive mixed-signal ATE, as well as the high test times for analog cores. Test methods for analog circuits that rely on low-cost digital testers are therefore especially desirable; a number of such methods have recently been developed [50, 51, 52, 53, 54, 55, 52].

Despite the numerous benefits of testing at the wafer level, industry practitioners have reported that mixed-signal test is seldom carried out at the wafer level [33, 56]. In our work, we present a new correlation-based signature analysis technique for mixed-signal cores, which facilitates defect screening at the wafer-level. The proposed technique is inspired by popular outlier analysis techniques for IDDQ testing [57, 58]. Outlier identification using IDDQ during wafer sort is difficult for deep-submicron processes [59]. This problem has been addressed using statistical post-processing techniques that utilize the test response data from the ATE [57]. We have developed a similar classification technique that allows us to make a pass/fail decision under non-ideal ambient conditions and using imprecise measurements. We present a wafer-scale analog test method based on the use of low-cost digital testers, and with reduced dependence on mixed-signal testers.

1.5 WLTBI of core-based SoCs

Several test scheduling techniques target reduction in overall test application time while considering power consumption constraints [16], precedence constraints during test [18], and conflicts between cores arising from the use of shared TAM wires. However, test scheduling for WLTBI has not thus far been addressed in research literature. In this thesis, we present a test scheduling technique that reduces the variation in power consumption during WLTBI.

1.6 Power management for WLTBI

The higher power consumption of ICs during scan-based testing is a serious concern in the semiconductor industry; scan power is often several times higher than the device power dissipation during normal circuit operation [60]. Excessive power consumption during scan testing can lead to yield loss. As a result, power minimization during test-pattern application has recently received much attention [61, 62, 63, 64, 65, 66]. Research has focused on pattern ordering to reduce test power [61, 67, 68]. The pattern-ordering problem has been mapped to the well-known Traveling Salesman Problem (TSP) [67, 68]. Testing semiconductor devices during burn-in at wafer-level requires low variation in power consumption during test [38]. A test-pattern reordering method that minimizes the dynamic power consumption does not address the needs of WLTBI. Specific techniques need to be developed to address this aspect of low-power testing testing, i.e., the ordering of test patterns to minimize the overall variation in power consumption.

In this thesis, we address the problem of power-conscious test-pattern ordering for WLTBI. The solutions methods, which are based on ILP and efficient heuristics, allow us to determine an appropriate ordering of test patterns that minimizes the overall cycle-by-cycle variation in power. Reduced variance in test power results in less fluctuations in the junction temperatures of the device. Test cubes generated by commercial automatic test pattern generation (ATPG) tools such as [69] have a significant percentage of don't-care bits in the test cubes. These unspecified bits in the test cubes can be filled with logic '0' and '1' to minimize peak/average power, enhance test compression, and increase the coverage of unmodeled defects. Several methods have been proposed to reduce the power consumption during scan testing by filling unspecified values in the test cubes [64, 70, 71]. In this thesis, we focus on a WLTBI-specific X-fill framework that can control the variation in power consumption during scan shift/capture.

1.7 Thesis outline

In this thesis we address three important practical problems: (i) wafer-level modular testing of core-based digital SoCs, (ii) wafer-level defect screening for "big-D/small-A" SoCs, and (iii) power management for WLTBI. These problems are solved with the underlying objective of lowering product cost, either by reducing the cost of packaging, or the cost of testing and the associated test infrastructure. The remainder of the thesis is organized as follows.

In Chapter 2, we present techniques for wafer-level modular testing of core-based SoCs. A statistical yield model is first developed to estimate the defect probabilities of the cores. This information is then used in an optimization framework to make decisions on the test-lengths for the cores under constraints of test application time. Similar techniques for wafer-level RPCT are also developed. Simulation results on the defect-screening probabilities are presented for five of the ITC'02 SoC Test benchmarks.

In Chapter 3, we propose a signature analysis technique for wafer-level defect-screening of "big-D/small-A" SoCs. A generic cost model is used to evaluate the effectiveness of wafer-level testing of analog and digital cores in a mixed-signal SoC, and to study its impact on test escapes, yield loss and packaging costs. Experimental results are presented for a typical "big-D/small-A" SoC, which contains a large section of flattened digital logic and several large mixed-signal cores.

A test-scheduling technique specifically suited for WLTBI of core-based SoCs is presented in Chapter 4. The primary objective of the test-scheduling technique is to minimize the variation in power consumption during test. A secondary objective is to minimize the test application time. The test-scheduling problem is modeled using multi-partite graphs and it is solved using a graph-matching technique. Simulation results are presented for three ITC02 SoC benchmarks, and the proposed technique is compared with two baseline methods.

In Chapter 5, we present a test-pattern ordering technique for WLTBI, where the objective is to minimize the variation in power consumption during test application. The test-pattern ordering problem for WLTBI is formulated and it is solved optimally using integer linear programming (ILP). Efficient heuristic methods are also presented to solve the pattern-ordering problem for large circuits. Simulation results are presented for the ISCAS'89 and the IWLS'05 benchmark circuits, and the proposed ordering technique is compared with two baseline methods that carry out pattern-ordering to minimize peak power and average power, respectively. A third baseline method that randomly orders test patterns is also used to evaluate the proposed methods.

In Chapter 6, we present a unified test-pattern manipulation and pattern-ordering technique for WLTBI, where the objective is to minimize the variation in power consumption during test application. Test-pattern manipulation is carried out by carefully filling the don't-care bits in test cubes. The X-fill problem is formulated and solved using an efficient polynomial-time algorithm. Simulation results are presented for the ISCAS'89 and the IWLS'05 benchmark circuits, and the proposed ordering technique is compared with three baseline methods that carry out pattern manipulation to minimize peak-power consumption as well as with a fourth baseline that targets only pattern compaction.

Finally, in Chapter 7, we present conclusions and identify future research directions.

Chapter 2

Test-Length Selection and TAM Optimization

In this chapter, we present an optimal test-length selection technique for wafer-level testing of core-based SoCs [72, 73]. This technique, which is based on a combination of statistical yield modeling and ILP, allows us to determine the number of patterns to use for each embedded core during wafer sort such that the probability of screening defective dies is maximized for a given upper limit on the SoC test time. Therefore, this work complements prior work on SoC test scheduling that lead to efficient test schedules that reduce the testing time during package test. For a given test access architecture, designed to minimize test time for all the scan patterns during package test, the proposed method can be used at wafer sort to screen defective dies, thereby reducing package cost and the subsequent test time for the IC lot. While an optimal test access architecture and test schedule can also be developed for wafer sort, we assume that these test planning problems are best tackled for package test, simply because the package test time is higher.

We also present an optimization framework for wafer sort that addresses TAM optimization and test-length selection for wafer-level testing of core-based digital SoCs when the tester has limited channel availability [74]. The objective here is to design a TAM architecture for that utilizes a pre-designed underlying TAM architecture for package test, and determine test-lengths for the embedded cores such that the overall SoC defect-screening probability at wafer sort is maximized. The proposed method reduces packaging cost and the subsequent test time for the IC lot, while efficiently utilizing available tester bandwidth at wafer sort.

The key contributions of this chapter are as follows:

- We show how statistical yield modeling for defect-tolerant circuits can be used to estimate defect probabilities for embedded cores in an SoC.
- We formulate the test-length selection problem for wafer-level testing of core-based SoCs. To the best of our knowledge, this is the first attempt to define a core-based test selection problem for SoC wafer sort.
- We develop an ILP model to obtain optimal solutions for the test-length selection problem. The optimal approach is applied to five ITC'02 SoC test benchmarks, including three from industry.
- We present an efficient heuristic approach to handle larger SoC benchmarks that may emerge in the near future.
- We present two techniques for test-length selection and TAM optimization. The first technique is based on the formulation of a non-linear integer programming model, which can be subsequently linearized and solved using standard ILP tools. While this approach leads to a thorough understanding of the optimization problem, it does not appear to be scalable for large SoCs. We therefore describe a second method that enumerates all possible valid TAM partitions, and then uses the ILP model presented in Section 2.2.1 to derive test-lengths to maximum defect screening at wafer sort. This enumerative procedure allows an efficient search of a large solution space. It results in significantly lower computation time than that needed for the first method. Simulation results on TAM optimization and test-length selection are presented for five of the ITC'02 SoC Test benchmarks.

2.1 Defect probability estimation for embedded cores

In this section, we show how defect probabilities for embedded cores in an SoC can be estimated using statistical yield modeling techniques.

2.1.1 Unified negative-binomial model for yield estimation

We adapt the yield model presented in [75, 76, 77] to model the yield of the individual cores in a generic core-based SoC. The model presented in [75] unifies the "small-area clustering" and "large-area clustering" models presented in [76] and [77], respectively. It is assumed in [75, 76] that the number of defects in a given area A is a random variable that follows a negative-binomial distribution. The negative binomial distribution is a two-parameter distribution characterized by the parameters λ_A and α_A. The parameter λ_A denotes the average number of defects in an area A. The clustering parameter α_A is a measure of the amount of defect clustering on the wafer. It can take values that range from 0.5 to 5 depending on the fabrication process, with lower values of α denoting increased defect clustering. The probability $\mathcal{P}(x, A)$ that x faults occur in area A is given by:

$$\mathcal{P}(x, A) = \frac{\Gamma(\alpha_A + x)}{x!\Gamma(\alpha_A)} \cdot \frac{(\lambda_A/\alpha_A)^x}{(1 + (\lambda_A/\alpha_A)^{\alpha_A + x}} \tag{2.1}$$

The above yield model was validated using industrial data in [78], and it has recently been used in [79, 80, 81]. An additional parameter incorporated in [75] is the *block size*, defined as the smallest value B such that the wafer can be divided into disjoint regions, each of size B, and these regions are statistically independent with respect to manufacturing defects. As in [75], we assume that the blocks are rectangular and can be represented by a tuple (B_1, B_2), corresponding to the dimensions of the rectangle. The goal of the yield model in [75] was to determine the effect of redundancy on yield in a fault-tolerant VLSI system. The basic redundant block is called a module, and the block is considered to be made up of an integer number of modules. Since our objective here is to model the yield of embedded (non-overlapping) cores in an SoC, we redefine the module to be an imaginary chip area denoted by (a_1, a_2). The size of the imaginary chip area, i.e., the values of a_1 and a_2 can be fixed depending on the resolution of the measurement system, e.g., an optical defect inspection setup. In this chapter we assume the dimensions of the imaginary chip area, a_1 and a_2, to be unity.

2.1.2 Procedure to determine core defect probabilities

We use the following steps to estimate the defect probabilities for the embedded cores:

(1) Determine the block size: Empirical data obtained on wafer maps and techniques described in [75] can be used to determine the block size. The block size helps us to determine the model parameters α_B and λ_B, where λ_B refers to the average number of defects within a block B of size (B_1, B_2), and α_B is the clustering parameter for the block. The size of the block plays an important role in our procedure to determine core defect probabilities. We next describe the procedure to determine the block size.

Efficient techniques for determining the block size have been presented in [75], and these techniques have been validated using empirical data. The block size can be determined using a simple iterative procedure, in which the wafer is divided into rectangular sub-areas (blocks), whose sizes are increased at every step. Starting with blocks of size $I = 1$, $J = 1$, we alternately increase I and J. For each fixed value of block size $I \times J$, we then calculate the corresponding parameter $\alpha_B(I, J)$ and arrange these values in a matrix. The value of (I, J), for which the difference between $\alpha_B(I, J)$

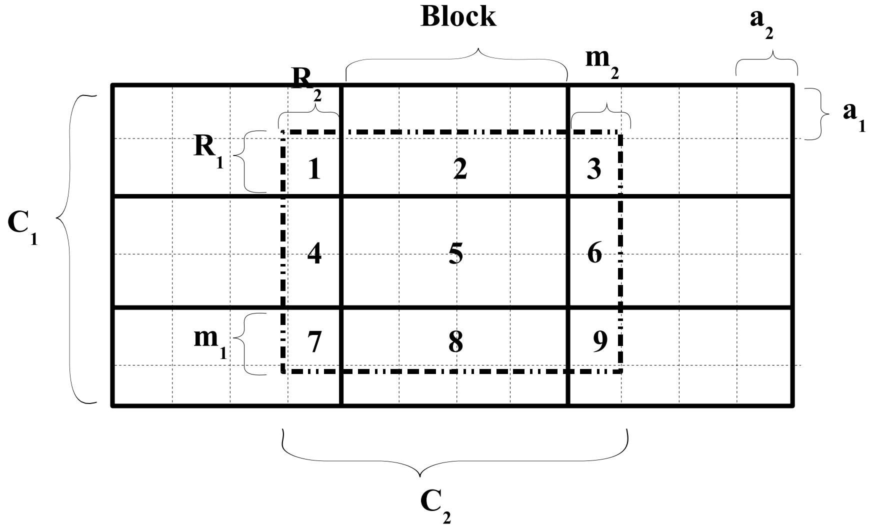


Figure 2.1: Defect estimation: Placement of a core with respect to blocks.

and $\alpha_B(1,1)$ is minimum, is chosen as the block size. The value of $\alpha_B(I,J)$ can be determined using standard estimation techniques such as the moment method, the maximum likelihood method, or curve fitting [76]. The clustering parameter remains constant within a block and increases when the are consists of multiple blocks [75, 76]; this property forms the basis for determining the block size.

In our work, we make the following assumptions: (a) as in [75], we assume that the area of the block consists of an integer number of imaginary chip areas; (b) the block size and its negative binomial parameters are pre-determined using rigorous statistical information processing of wafer defect maps. The illustration in Figure 2.1 represents a cross section of a wafer and its division among blocks. The dimensions of the block in Figure 2.1 is $(2,3)$, and each block contains 8 imaginary chips of area $(1,1)$.

(2) We consider each core in the SoC to be an "independent chip". Let us consider a core represented by (C_1,C_2), block size (B_1,B_2), and imaginary chip (a_1,a_2). The imaginary chip is a sub-area in a block. For a fault in a block, the distribution of the fault within the area of the block is uniform; the imaginary chip area parameters

λ_m and α_m take on values λ_B/B and α_B respectively. The relationship between the imaginary chip area parameters and the block parameters can be established using techniques proposed in [75]. The purpose of dividing a wafer into blocks, is to facilitate the division of a wafer into sub-areas, such that distinct fault clusters are contained in distinct blocks (each block is statistically independent with respect to manufacturing defects). We now determine the probability that the core is defective using the following steps:

(a) In a statistical sample of multiple wafers, a core can be oriented in different configurations with respect to the block. The number of possible orientations of the core with respect to the block in the wafer is given by $\min\{B_1, C_1\} \times \min\{B_2, C_2\}$. The dimensions of the block in Figure 2.1 are smaller than that of the core. The number of possible orientations for the core in Figure 2.1 is therefore 2×4, i.e., there are 8 possible core orientations with respect to the block in Figure 2.1. The list of possible values (R_1, R_2) in Figure 2.1 can take, (1,1), (1,2), (1,3), (1,4), (2,1), (2,2), (2,3), (2,4), intuitively illustrates the 8 possible core orientations with respect to a block of size (2,4).

(b) For each orientation, determine the distance from the top-left corner of the core to the closest block boundaries. This is represented as (R_1, R_2), the two values denoting distances in the Y and X directions, respectively; the placement of the core with respect to the block determines the way the core is divided into complete and partial blocks. In Figure 2.1, we have $R_1 = R_2 = 1$.

(c) The dimensions of the core can now be represented as $C_1 = R_1 + n_1 \cdot B_1 + m_1$, and $C_2 = R_2 + n_2 \cdot B_2 + m_2$, where n_1 and m_1 are defined as:

$$n_1 = \lfloor (\tfrac{C_1 - R_1}{B_1}) \rfloor$$
$$m_1 = (C_1 - R_1) \mod B_1$$

The parameters n_2 and m_2 are defined in a similar fashion. The values of $n_1, m_1, n_2,$

m_2 for the illustrated orientation in Figure 2.1 are all 1.

(d) The core can be divided into a maximum of nine disjoint sub-areas for the orientation illustrated in Figure 2.1, with each sub-area placed in a different block. Dividing the core into independent sub-areas allows for the convolution of the probability of failure of each individual sub-area. Let us assume that there are a total of D sub-areas; the probability that the core is defect-free is given by $\mathcal{P}^{(R_1,R_2)} = \prod_{i=1}^{D} a(N_i)$. The superscript (R_1, R_2) indicates the dependence of this probability on the placement. Here $a(N_i)$ denotes the probability that all the N_i imaginary chip areas in the sub-area i are defect-free. This probability can be obtained from Equation (2.2) shown below, where $a(k, N)$ denotes the probability of k defect-free modules in a sub-area with N modules. By substituting N instead of k in Equation (2.2), we obtain Equation (2.3). This is done in order to estimate the probability that a block is fault-free.

$$a(k, N) = \binom{N}{k} \sum_{i=0}^{N-k} (-1)^i \binom{N-k}{i} \left(1 + \frac{(i+k)\lambda_m}{\alpha_m}\right)^{-\alpha_m} \tag{2.2}$$

$$a(N, N) = a(N) = \left(1 + \frac{N\lambda_m}{\alpha_m}\right)^{-\alpha_m} \tag{2.3}$$

The process of dividing the area of a core into multiple sub-areas facilitates the application of large-area clustering conditions on the individual sub-areas. It is important to distinguish between sub-areas $i = 1, 3, 7, 9$ and $i = 2, 4, 5, 6, 8$ in Figure 2.1. In the latter case, the sub-area i is divided into several parts, each contained in a different block. The derivation of the probability density function for these sub-areas is now a trivial extension of the base case represented by Equation (2.3).

(e) The final step is the estimation of the defect probability for the core. We first estimate the probability that the core is defect-free for all possible values of R_1 and

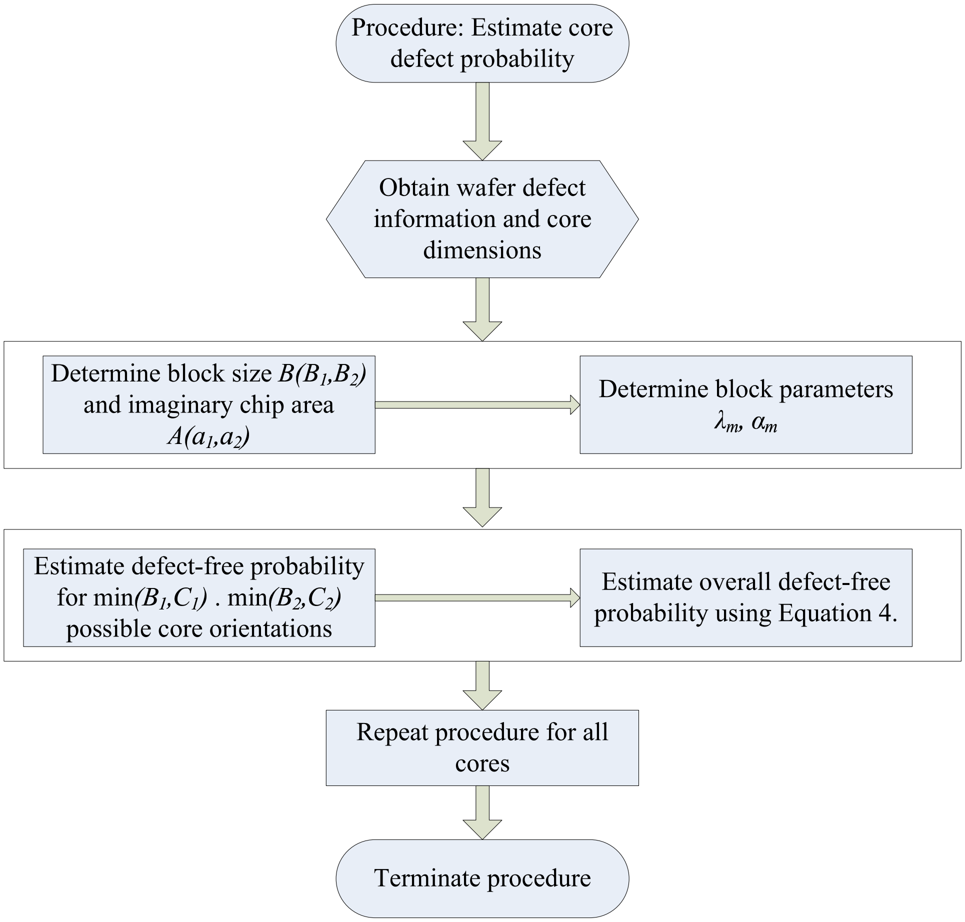


Figure 2.2: Flowchart depicting the sequence of procedures used to estimate core defect probabilities.

R_2. The overall defect-free probability P is obtained by averaging the defect-free probability over all orientations, and it is given by:

$$\mathcal{P} = \frac{1}{\min(B_1, C_1) \cdot \min(B_2, C_2)} \sum_{R_1=1}^{\min(B_1,C_1)} \sum_{R_2=1}^{\min(B_2,C_2)} \mathcal{P}^{(R_1,R_2)} \tag{2.4}$$

We use Figure 2.1 to illustrate the calculation of the defect probability for an embedded core. The figure represents the relative placement of a core with respect to the blocks. We have a block size of $(4, 2)$, a core size of $(6, 4)$ and imaginary chip area of size $(1, 1)$. The core is divided into nine distinct sub-areas numbered $1-9$.

For values of $\alpha_B = 0.25$ and $\lambda_B = 0.1$, we now determine the probability that the core is defect-free using Equation (2.3):

$$\begin{aligned}
\mathcal{P}^{(1,1)} &= a(R_1 \cdot R_2) \cdot a(R_1 \cdot B_2)^{n_2} \cdot a(R_1 \cdot m_2) \cdot \\
&\quad a(B_1 \cdot R_2)^{n_1} \cdot a(B_1 \cdot B_2)^{n_1 \times n_2} \cdot a(B_1 \cdot m_2) \cdot \\
&\quad a(m_1 \cdot R_2) \cdot a(m_1 \cdot B_2)^{n_2} \cdot a(m_1 \cdot m_2) \\
&= a(1) \cdot a(4)^{n_2} \cdot a(1) \cdot a(2)^{n_1} \cdot a(8)^{n_1 \times n_2} \cdot a(2) \cdot \\
&\quad a(1) \cdot a(4)^{n_2} \cdot a(1) \\
&= (0.9879) \cdot (0.9554) \cdot (0.9879) \cdot (0.9765) \cdot (0.9193) \\
&\quad \cdot (0.9765) \cdot (0.9879) \cdot (0.9879) \cdot (0.9554) \\
&= 0.76206 \qquad (2.5)
\end{aligned}$$

The above procedure is repeated until the defect-free probability for all $\min(B_1, C_1) \times \min(B_2, C_2)$ combinations of R_1 and R_2 are determined. The final core defect-free probability is then calculated using Equation (2.4). The probability that the core has a defect is simply $\bar{\mathcal{P}} = 1 - \mathcal{P}$. For a given SoC, this procedure can be repeated for every embedded core until all defect probabilities are obtained. The flowchart in Figure 2.2 summarizes the sequence of procedures that lead to the estimation of core defect probabilities. The procedure begins by accumulating wafer defect information and information on the individual core dimensions. This information is then used to determine the size of the block and the block parameters, λ_b and α_b. These are then used to calculate parameters for the imaginary chip area. The defect probability of the core is then calculated for all possible core orientations, with respect to a block in the wafer; the defect probability of the core is the calculated using Equation (2.4).

The knowledge of the dimensions of each individual core is necessary to determine the corresponding defect probabilities. In this chapter, we use the overall SoC dimensions as given in [82] to derive information pertaining to the size of the individual modules in the ITC'02 SOC Test benchmarks. Since these benchmarks do not provide information about the sizes of the embedded cores, we use the total number of patterns for each core as an indicator of size. This assumption helps us extract the relative size of a core by normalizing it with respect to the overall SoC dimensions. We use layout information in the form of $X-Y$ coordinates for the SoC as described in [82]; the bottom-left corner of the SoC has $X-Y$ coordinates of $(0,0)$, and the layout information provides information on the $X-Y$ coordinates of the top-right corner of the SoC. The sequence of procedures in Figure 2.2 is then performed to determine the core defect probabilities. Table 2.1 shows the defect probabilities for each core in four of the ITC'02 SoC test benchmark circuits [83], estimated using the parameters $\alpha_B = 0.25$ and $\lambda_B = 0.035$.

2.2 Test-length selection for wafer-level test

In this section, we formulate the problem of determining the test-length for each embedded core, such that for a given upper limit on the SoC test time (expressed as a percentage of the total SoC test time), the defect-screening probability is maximized. We present a framework that incorporates the defect probabilities of the embedded cores in the SoC, the upper bound on SoC test time at wafer sort, the test lengths for the cores, and the probability that a defective SoC is screened. The defect probabilities for the cores are obtained using the yield model presented in Section 2. Let us now define the following statistical events for Core i:

A_i: the event that the core has a fault; the probability associated with this event is determined from the statistical yield model.

B_i: the event that the tests applied to Core i do not produce an incorrect response. $\bar{A}_i$ and $\bar{B}_i$ represent events that are complementary to events A_i and B_i, respectively.

Core Number	Defect Probability			
	SoC: d695	SoC: p34392	SoC: p22810	SoC: p93791
1	0.0038	0.1126	0.9864	0.5825
2	0.1104	0.4597	0.9234	0.1756
3	0.1160	0.8821	0.9986	0.8886
4	0.2162	0.7983	0.9931	0.0006
5	0.2339	0.6841	0.9837	0.9926
6	0.6946	0.9997	0.9132	0.2202
7	0.1766	0.4552	0.9998	0.1508
8	0.1882	0.9642	0.9060	0.1508
9	0.0038	0.1309	0.9545	0.1756
10	0.0960	0.3877	0.1356	0.9992
11		0.9192	0.5904	0.1678
12		0.0824	0.5820	0.5501
13		0.9010	0.0002	0.1788
14		0.3609	0.6920	0.1788
15		0.9118	0.1294	0.3515
16		0.0503	0.0065	0.5596
17		0.7469	0.0615	0.2167
18		0.7134	0.9982	0.0092
19		0.9432	0.2887	0.2061
20			0.7889	0.5943
21			0.9991	0.0092
22			0.2964	0.0092
23			0.1490	0.2491
24			0.0712	0.9789
25			0.9906	0.9981
26			0.9633	0.0469
27			0.0004	0.9875
28			0.0666	0.5596
29				0.1427
30				0.1756
31				0.1956
32				0.9110

Table 2.1: Core defect probabilities for four ITC'02 SoC test benchmark circuits.

Two important conditional probabilities associated with the above events are yield loss and test escape, denoted by $\mathcal{P}(\bar{B}_i \mid \bar{A}_i)$ and $\mathcal{P}(B_i \mid A_i)$, respectively. Using a basic identity of probability theory, we can derive the probability that the test applied to Core i detects a defect:

$$\mathcal{P}(\bar{B}_i) = \mathcal{P}(\bar{B}_i \mid A_i) \cdot \mathcal{P}(A_i) + \mathcal{P}(\bar{B}_i \mid \bar{A}_i) \cdot \mathcal{P}(\bar{A}_i) \tag{2.6}$$

Due to SoC test time constraints during wafer-level testing, only a subset of the pattern set can be applied to any Core i, i.e., if the complete test suite for the SoC contains p_i scan patterns for Core i, only $p_i^* \leq p_i$ patterns can be actually applied to it during wafer sort. Let us suppose the difference between the SoC package test time and the upper limit on wafer sort test time is ΔT clock cycles. The test time for each TAM partition therefore needs to be reduced by ΔT clock cycles, if we assume that the package test times on the TAM partitions are equal. The value of p_i^* adopted for Core i depends on its wrapper design. The larger the difference between the external TAM width and internal test bitwidth (number of scan chains plus the number of I/Os), the greater the impact of $(p_i - p_i^*)$ on ΔT. In fact, given two cores (Core i and Core j) with different wrapper designs, the reduction in the number of patterns by the same amount, i.e., $p_i - p_i^* = p_j - p_j^*$, can lead to different amount of reductions in core test time (measured in clock cycles). Let $fc_i(p_i^*)$ be the fault coverage for Core i with p_i^* test patterns.

We next develop the objective function for the test-length selection problem. The goal of this objective function is to satisfy two objectives: (1) Maximize the probability that Core i fails the test; 2) Minimize the overall test escape probability. The ideal problem formulation is one that leads to an objective function satisfying both the above objectives.

Let us now assume that the yield loss is γ_i, the test escape is β_i, and the probability that Core i has a defect is θ_i. Using these variables, we can rewrite Equation (2.6) as:

$$\mathcal{P}(\bar{B}_i) = fc_i(p_i^*) \cdot \theta_i + \gamma_i \cdot (1 - \theta_i) \tag{2.7}$$

Similarly we can rewrite $\mathcal{P}(B_i)$ as follows:

$$\mathcal{P}(B_i) = 1 - \mathcal{P}(\bar{B_i}) = \theta_i \cdot \beta_i + (1 - \gamma_i) \cdot (1 - \theta_i) \tag{2.8}$$

We therefore conclude that for a given value of α_i and γ_i, the objective function that maximizes the probability $\mathcal{P}(\bar{B_i})$ that Core i fails the test, also minimizes the test escape β_i. Therefore, it is sufficient to maximize $\mathcal{P}(B_i)$ to ensure that the test escape rate is minimized. In our study, we assume that the yield loss γ_i is negligible for each core. Assuming that the cores fail independently with the probabilities derived in Section 2, the defect-screening probability $\mathcal{P}_S$ for an SoC with N embedded cores is given by $\mathcal{P}_S = 1 - \prod_{i=1}^{N} \mathcal{P}(B_i)$.

2.2.1 Test-length selection problem: $\mathscr{P}_{TLS}$

We next present the test-length selection problem $\mathscr{P}_{TLS}$, wherein we determine an optimal number of test patterns for each core in the SoC, such that we maximize the probability of screening defective dies at wafer sort for a given upper limit on the SoC test time. We assume a fixed-width TAM architecture as in [8], where the division of W wires into $\mathbf{B}$ TAM partitions, and the assignment of cores to the $\mathbf{B}$ TAM partitions have been determined *a priori* using methods described in [84, 8, 85, 10].

Let the upper limit on the test time for an SoC at wafer sort be T_{max} (clock cycles). This upper limit on the scan test time at wafer sort is expected to be a fraction of the scan test time T_{SoC} (clock cycles) for package test, as determined by the TAM architecture and test schedule. The fixed-width TAM architecture requires that the total test time on each TAM partition must not exceed T_{max}.

If the internal details of the embedded cores are available to the system integrator, fault simulation can be used to determine the fault coverage for various values of p_i^*, i.e., the number of patterns applied to the cores during wafer sort. Otherwise, we model the relationship between fault coverage and the number of patterns with an exponential function. It is well known in the testing literature that the fault coverage for stuck-at faults increases rapidly initially as the pattern count increases, but it flattens out when more patterns are applied to the circuit under test [2, 86]. In our work, without loss of generality, we use the normalized function $fc_i(p_i^*) = \frac{\log_{10}(p_i^*+1)}{\log_{10} p_i}$ to represent this relationship. A similar relationship was used in [86]. We have verified that this empirical relationship matches the "fault coverage curve" for the ISCAS benchmark circuits.

Let $\epsilon_i(p_i^*)$ be the defect-escape probability for Core i when p_i^* patterns are applied to it. This probability can be obtained using Equation (2.8) as a function of the test escape β_i and the probability θ_i that the core is faulty. The value of θ_i for each core in the SoC is obtained using the procedure described in Section 2.2.

The optimization problem $\mathscr{P}_{TLS}$ can now be formally stated as follows:

$\mathscr{P}_{TLS}$: Given a TAM architecture for a core-based SoC and an upper limit on the SoC test time, determine the total number of test patterns to be applied to each core such that: (i) the overall testing time on each TAM partition does not exceed the upper bound T_{max} and (ii) the defect-screening probability $\mathcal{P}(\bar{B}_i)$ for the SoC is maximized. The objective function for the optimization problem is as follows:

$$\text{Maximize } Y = \prod_{i=1}^{N} 1 - \mathcal{P}(B_i)$$

where the number of cores in the SoC is N. We next introduce the indicator binary variable δ_{ij}, $1 \leq i \leq N$, $0 \leq j \leq p_i$, which ensure that exactly one test-length is selected for each core. It is defined as follows:

$$\delta_{ij} = \begin{cases} 1 & \text{if } p_i^* = j \\ 0 & \text{otherwise} \end{cases}$$

where $\sum_{i=1}^{p_i} \delta_{ij} = 1$. The defect escape probability ϵ_i^* for Core i is given by $\epsilon_i^* = \sum_{j=1}^{q_i} \delta_{ij}\epsilon_i(j)$. We next reformulate the objective function to make it more amenable for further analysis. Let $\mathcal{F} = \ln(Y)$. We therefore get:

$$\begin{aligned}
\mathcal{F} &= \ln(Y) \\
&= \ln\big(\prod_{i=1}^{N} 1 - \mathcal{P}(B_i)\big) \\
&= \sum_{i=1}^{N} \ln(1-\epsilon_i) \\
&= \sum_{i=1}^{N} \ln\Big(1 - \sum_{j=1}^{p_i} \delta_{ij}\epsilon_i(j)\Big)
\end{aligned}$$

We next use the Taylor series expansion $\ln(1-x) = -\Bigg(x + \frac{x^2}{2} + \frac{x^3}{3} + \cdots\Bigg)$ and ignore the second- and higher-order terms [87]. This approximation is justified if the defect-escape probability for Core i is much smaller than one. While this is usually the case, occasionally the defect-escape probability is large; in such cases, the optimality claim is valid only in a limited sense. The impact that this approximation has on the overall defect-screening probability of the SoC is examined in Section 2.3. The simplified objective function is given by:

$$\text{Maximize } \mathcal{F} = \sum_{i=1}^{N} \Bigg(\sum_{j=1}^{p_i} -\Big(\delta_{ij}\epsilon_i(j)\Big)\Bigg) \tag{2.9}$$

In other words, our objective function can be stated as

$$\text{Minimize } \mathcal{F} = \sum_{i=1}^{N} \Bigg(\sum_{j=1}^{p_i} \delta_{ij}\epsilon_i(j)\Bigg) \tag{2.10}$$

Next we determine the constraints imposed by the upper limit on the SoC test time. Suppose the SoC-level TAM architecture consists of **B** TAM partitions. Let $T_i(j)$ be the test time for Core i when j patterns are applied to it. For a given Core i on a TAM partition of width w_B, we use the design-wrapper technique from [8] to determine the longest scan in (out) chains of length $s_i(s_o)$ of the core on that TAM partition. The value of $T_i(j)$ can be determined using the formula $T_i(j) = (1 + \max\{s_i, s_o\} \cdot j + \min\{s_i, s_o\})$ [8]. The test time T_i^* for Core i is therefore given by $T_i^* = \sum_{j=1}^{p_i} \delta_{ij} T_i(j)$. Let A_j denote the set of cores that are assigned to TAM partition j. We must ensure that $\sum_{Core_i \in A_j} T_i^* \leq T_{max}, 1 \leq j \leq B$.

The number of variables and constraints for a given ILP model determines the complexity of the problem. The number of variables in the ILP model is only $N + N\sum_{i=1}^{N} p_i$, and the number of constraints is only $\sum_{i=1}^{N} N \cdot p_i + B$; thus this exact approach is scalable for large problem instances. The complete ILP model is shown as Figure 2.3.

Minimize $\mathcal{F} = \sum_{i=1}^{N} \left(\sum_{j=1}^{p_i} \delta_{ij} \epsilon_i(j) \right)$ subject to:

1. $\sum_{i=1}^{p_i} \delta_{ij} = 1,\ 1 \leq i \leq N,\ 0 \leq j \leq p_i$
2. $\sum_{\text{Core}_i \in A_j} T_i^* \leq T_{max}, 1 \leq j \leq B$
3. $\delta_{ij} = 0$ or $1,\ 1 \leq i \leq N,\ 0 \leq j \leq p_i$

/* Constants : $\epsilon_i(j),\ T_i(j)$ */

/* Variables : $\delta_{ij},\ T_i^*,\ 1 \leq i \leq N,\ 0 \leq j \leq p_i$ */

Figure 2.3: Integer linear programming model for $\mathscr{P}_{TLS}$.

2.2.2 Efficient heuristic procedure

The exact optimization method based on ILP is feasible for the largest benchmarks (contributed by Philips) in the ITC'02 SoC benchmark set. While these three benchmarks are representative of industrial designs in 2002, current core-based SoCs are larger in size. To handle such SoCs, we present a heuristic approach to determine the test-length p_i^* for each core, given the upper limit on maximum SoC test time. The heuristic method consists of a sequence of five procedures. The objective of the heuristic method is similar to that for the ILP technique, i.e., to maximize the overall defect-screening probability. The heuristic method performs an iterative search over the TAM partitions. In each, step we identify a core for which a reduction in the number of applied patterns results in a minimal decrease in the overall defect-screening probability. This procedure is repeated until the time constraint on all TAM partitions is satisfied. We next describe the procedures that make up the heuristic method.

1. We begin our heuristic procedure by assuming that all patterns are applied to each core. This assumption implies that $\sum_{Core_i \in A_j} T_i^* = T_{SoC}, 1 \leq j \leq B$.

2. In procedure T_{pat}_*Reduce*, for each TAM partition $j, 1 \leq j \leq B$, we chose a particular $Core_i \in A_j$ such that a decrease in the number of applied patterns Δp_i^* results in a minimal decrease in $\epsilon_i(p_i^*)$; we consider different values of Δp_i^* in the range $1 \leq \Delta p_i^* \leq 15$ in our experiments, and choose the value that results in maximum defect screening for the SoC. Procedure T_{pat}_*Reduce*, searches for a core in each TAM partition, which yields a maximum value for $\theta_i \cdot (fc_i(p_i^*) - (fc_i(p_i^* - \Delta p_i^*)$. For the sake of simplicity, we assume that the yield loss γ_i is negligible for each core.

3. We use the design wrapper technique in our next procedure step, T_{time}_*Update*,

to determine the test time reduction for Core i (obtained using T_{pat}_*Reduce*), corresponding to the reduction in the number of test patterns Δp_i^*. We denote ΔT_{ij} as the reduction in test time obtained by reducing the number of test patterns for Core i on TAM partition j by Δp_i^*; this can be obtained by solving the following equation: $\Delta T_{max_{ij}} = \lceil (1 + \max(s_i, s_o)_{ij}) \cdot \Delta p_i^* + \min(s_i, s_o)_{ij} \rceil$. The core test time, T_i^*, is now updated as $T_i^* - \Delta T_{max_{ij}}$.

4. The T_{max}_*Check* procedure checks whether $\sum_{Core_i \in A_j} T_i^* \leq T_{max}, 1 \leq j \leq B$. This procedure is performed each time after procedure T_{pat}_*Reduce* is executed.

5. If the check in procedure T_{max}_*Check* returns true for all TAM partitions, we then compute the overall defect-screening probability for the SoC.

A sort operation is performed each time procedure T_{pat}_*Reduce* is executed. Hence the worst-case computational complexity of the heuristic procedure is O($p_{_T} \cdot N \log N$), where N is the number of cores in the SoC, and $p_{_T} = \sum_{i=1}^{N} p_i$ is the total number of test patterns for package test for all the cores. The pseudocode for the heuristic procedure, as shown in Algorithm 1, calculates the test-lengths and the defect-escape probabilities for each core in the SoC.

2.2.3 Greedy heuristic procedure

We now present a greedy heuristic procedure to solve the test-length selection problem. This procedure was developed to demonstrate the need for an iterative heuristic procedure that reduces the core test-lengths with minimal impact on defect-screening. The heuristic approach in this section determines the test length p_i^* for each core, given the upper limit on maximum SoC test time as a constraint. Let us suppose there are **B** TAM partitions in the SoC test access architecture. It is obvious that

Algorithm 1 Test-Length Selection

```
Let $T_{max}$ be the constraint on wafer test time for the SoC, $T_{max} = k \cdot T_{SoC}$, $0 \leq k \leq 1$;
Let $\mathbf{B}$ = total number of TAM partitions;
Let $k$ = fraction of $T_{SoC}$ permissible for wafer test;
$\sum_{Core_i \in A_j} T_i^* = T_{SoC}, 1 \leq j \leq B$;
while time constraint for the SoC is not satisfied for TAM partition $j$, $1 \leq j \leq \mathbf{B}$ do
    for all cores in $A_j$ do
        Find $i$ such that $\theta_i \cdot (fc_i(p_i^*) - (fc_i(p_i^* - \Delta p_i^*)$ is maximum;
    end for
    $\Delta T_{max_{ij}} = \lceil (1 + \max(s_i, s_o)_{ij}) \cdot \Delta p_i^* + \min(s_i, s_o)_{ij} \rceil$;
    $T_i^* = T_i^* - \Delta T_{max_{ij}}$;
    for all TAM partitions, $1 \leq j \leq \mathbf{B}$ do
        if $\sum_{Core_i \in A_j} T_i^* \leq T_{max}, 1 \leq j \leq B$ then
            Compute relative defect-screening probability for the SoC;
        end if
    end for
end while
return relative defect-screening probability $\mathcal{P}_S^r$ for the SoC;
```

we can satisfy the constraint on T_{max} if we reduce the test time for all the cores in each TAM partition to a fraction of the original test time.

Let us denote the maximum wafer-test time for Core i on TAM partition j as $T_{max_{ij}}$. The test-length for the core corresponding to the test time $T_{max_{ij}}$ is given by $p_i^* = \lfloor \frac{T_{max_{ij}} - \min(s_i, s_o)_{ij}}{1 + \max(s_i, s_o)_{ij}} \rfloor$. With the knowledge of the test-length p_i^* for each core in the SoC, we can then proceed to determine the corresponding defect-escape probabilities $\epsilon_i(p_i^*)$, and then the overall defect-escape probability of the SoC given by $\epsilon_{SoC} = \prod_{i=1}^{N} \left(1 - \epsilon_i(p_i^*)\right)$. The heuristic procedure is simple and has a computational complexity of only $O(N)$. The above procedure is reasonable if the test times on the TAM partitions are fairly close to one another. This however is not the case in most industrial designs because of the heterogeneous nature of the cores in the SoC. The pseudocode for the heuristic procedure, which calculates the test-lengths and the defect-escape probabilities for each core in the SoC, is shown in Algorithm 2.

Algorithm 2 Test-length selection

```
Let $T_{max} = k \cdot T_{SoC}$, $0 \leq k \leq 1$; /*Constraint on wafer-test time for the SoC*/
Let $\mathbf{B}$ = total number of TAM partitions;
Let $k$ = fraction of $T_{SoC}$ permissible for wafer test;
Let $\epsilon_{SoC}$ = overall SoC defect escape probability during wafer-test;
while all core test-lengths have not been determined, do
    for $TAM_j \leftarrow 1$ to B do
        Calculate $\max(s_i, s_o)_{ij}$ and $\min(s_i, s_o)_{ij}$, $\forall_i$ on $TAM_j$;
        $p_i^* = \lfloor \frac{T_{max_{ij}} - \min(s_i,s_o)_{ij}}{1+\max(s_i,s_o)_{ij}} \rfloor$, $\forall_i$ on $TAM_j$ ;
        Calculate $\epsilon_i(p_i^*)$, $\forall_i$ on $TAM_j$ ;
    end for
end while
$\epsilon_{SoC} = \prod_{i=1}^{N} \Big(1 - \epsilon_i(p_i^*)\Big)$;
```

2.3 Experimental results

In this section, we present experimental results for five SoCs from the ITC'02 SoC test benchmark suite [83]. We use the public domain ILP solver *lpsolve* for our experiments [88]. Since the objectives of our experiment are to select the number of test patterns in a time-constrained wafer sort test environment, and at the same time maximize the defect-screening probability for the SoC, we present the following results:

- Given values of W and T_{max} relative to T_{SoC}, the percentage of test patterns for each individual core that must be applied at wafer sort to maximize the defect-screening probability for the SoC.

- The relative defect-screening probability $\mathcal{P}_S^r$ for each core in an SoC, where $\mathcal{P}_S^r = \mathcal{P}_S/\mathcal{P}_S^{100}$ and $\mathcal{P}_S^{100}$ is the defect-screening probability if all 100% of the patterns are applied per core.

- The relative defect-screening probability for each SoC obtained using the ILP model and the proposed heuristic methods.

- Approximation errors in $\mathcal{P}_S^r$ due to the Taylor series approximation.

Table 2.2: Defect screening probabilities: ILP-based approach versus proposed heuristic approaches.

SoC	W	$T_{max} = 0.75\ T_{SoC}$			$T_{max} = 0.5\ T_{SoC}$			$T_{max} = 0.25\ T_{SoC}$		
		Optimal Method	Heuristic Method	Greedy Method	Optimal Method	Heuristic Method	Greedy Method	Optimal Method	Heuristic Method	Greedy Method
d695	8	0.9229	0.7316	0.4111	0.6487	0.5343	0.1091	0.4095	0.3834	0.0039
	16	0.9229	0.7316	0.4113	0.6487	0.5759	0.1091	0.4308	0.3706	0.0039
	24	0.9047	0.6400	0.4110	0.5985	0.3106	0.1091	0.3604	0.1779	0.0039
	32	0.8765	0.4627	0.4110	0.5245	0.4024	0.1091	0.1666	0.1088	0.0039
p22810	8	0.7693	0.7473	0.1563	0.5947	0.4763	0.0053	0.0969	0.0302	~ 0
	16	0.8137	0.6994	0.1553	0.5996	0.3302	0.0047	0.1699	0.0524	~ 0
	24	0.7871	0.7079	0.0966	0.3340	0.3190	0.0032	0.0143	0.0012	~ 0
	32	0.7656	0.5736	0.1553	0.3435	0.1706	0.0032	0.0414	0.0005	~ 0
p34392	8	0.8661	0.6576	0.0042	0.6869	0.3513	~ 0	0.3521	0.1036	~ 0
	16	0.8807	0.6965	0.0041	0.7118	0.4400	~ 0	0.2157	0.0780	~ 0
	24	0.8990	0.7010	0.0042	0.7207	0.4315	~ 0	0.2569	0.1835	~ 0
	32	0.9161	0.5783	0.0042	0.6715	0.3007	~ 0	0.2278	0.0833	~ 0
p93791	8	0.4883	0.4406	0.1539	0.2299	0.0716	0.0097	0.0097	0.0048	~ 0
	16	0.5341	0.4420	0.1539	0.2438	0.1161	0.0097	0.0168	0.0088	~ 0
	24	0.7234	0.5547	0.1539	0.2535	0.1354	0.0096	0.0826	0.0015	~ 0
	32	0.7098	0.6317	0.1539	0.3335	0.1351	0.0097	0.0548	0.0037	~ 0

We first present results on the number of patterns determined for the cores. The results are presented in Figures 2.4-2.6 for three values of T_{max}: $0.75T_{SoC}$, $0.50T_{SoC}$, and $0.25T_{SoC}$. For the three large "p" SoCs from Philips, we select the value of $\mathbf{B}$ that minimizes the SoC package test time. The results show that the fraction of patterns applied per core, while close to 100% in many cases, varies significantly in order to maximize the SoC defect-screening probability. The maximum value of TAM width W (in bits) is set to 32 and we repeat the optimization procedure for all TAM widths ranging from 8 to 32 in steps of eight. Results are reported only for $W = 8$; similar results are obtained for other values of W. The CPU time for *lpsolve* for the largest SoC benchmark was less than a second.

We next present the defect-screening probabilities for all the individual cores in the benchmark SoCs (Figures 2.7-2.9). The cores that are more likely to lead to fails during wafer sort exhibit higher defect-screening probabilities, and vice versa. A core with small defect probability ends up having more patterns removed from the initial test suite during wafer sort. This is because a manufacturing defect is unlikely to cause a failure in that core. The second reason for low relative defect-screening probability is because certain cores have very few patterns that need to be applied when test-lengths are reduced for these cores. As a result, we obtain significantly low relative defect-screening probabilities for these cores. Even though the large SoCs have low relative defect-screening probabilities, these are the optimal values under the given test time constraints at wafer sort.

Finally, we compare our ILP-based optimization technique with the two heuristic procedures on the basis of relative SoC defect-screening probabilities obtained using the two methods. The values of the defect-screening probabilities $\mathcal{P}_S$ of the benchmark SoCs obtained using both the ILP-based model and the heuristic method for varying TAM widths, as well as overall test time are summarized in Table 2.2.

The results show that, as expected, the ILP-based method leads to higher defect-screening probabilities when compared with the heuristic procedure. Nevertheless, the heuristic procedure is efficient for defect screening when $T_{max} = 0.75T_{SoC}$ and $0.5T_{SoC}$. The greedy heuristic method on the other hand yields poor defect-screening probabilities compared to the ILP method and the heuristic method. This shows that the proposed heuristic method is effective for screening dies at wafer sort testing of large SoCs. A significant percentage of the faulty dies can be screened at wafer sort using our proposed techniques.

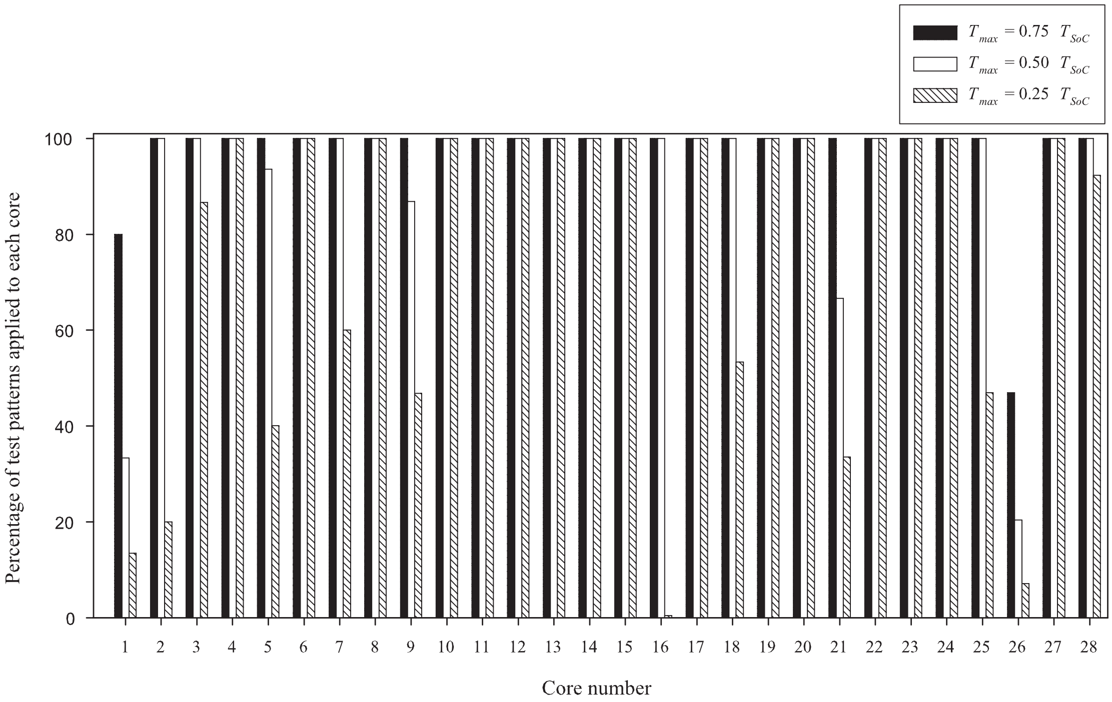


Figure 2.4: Percentage of test patterns applied to each core in p22810 for $W = 8$.

Approximation error in $\mathcal{P}_S^r$ due to Taylor series approximation

A Taylor's series expansion of $\delta_i(j)\epsilon_i(j)$, without the higher-order terms, was used in Section 2.2 to obtain a linear objective function for $\mathcal{P}_{TLS}$. If the defect-escape probability for Core i is much smaller than unity, this assumption can be justified.

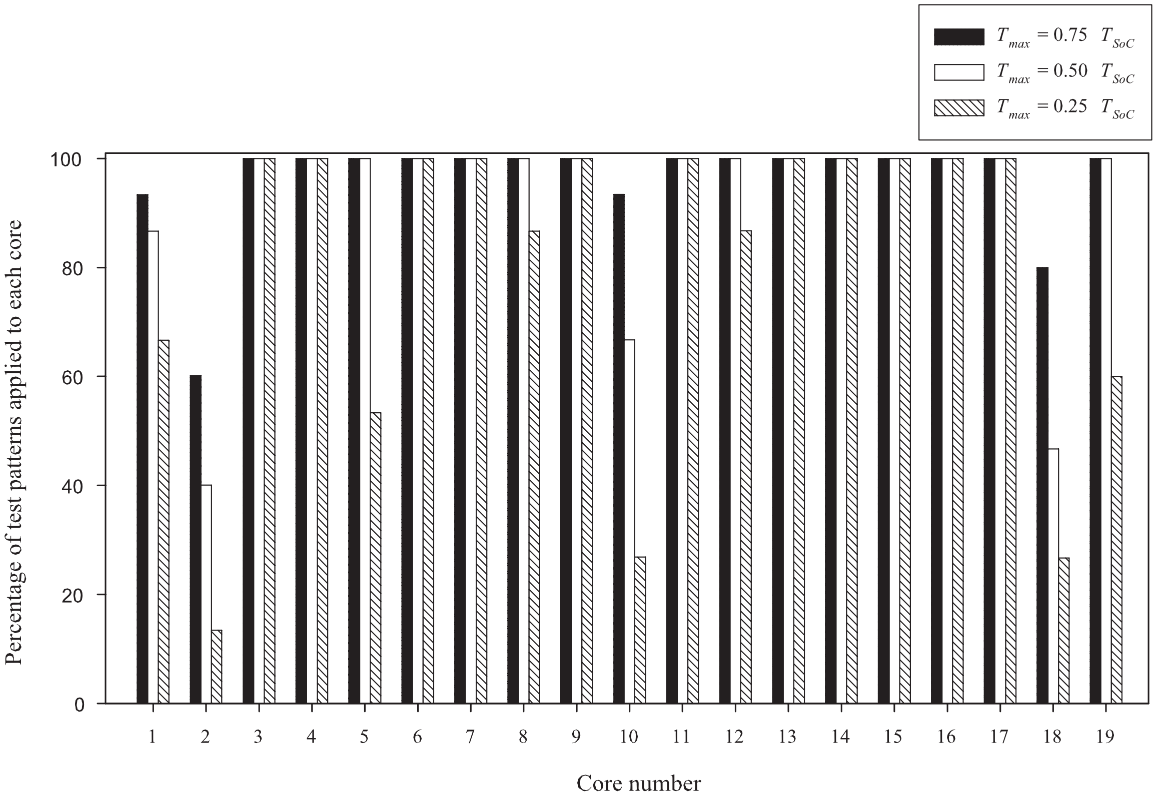


Figure 2.5: Percentage of test patterns applied to each core in p34392 for $W = 8$.

To study the effect of this approximation, we evaluated the approximation error for industrial designs. We used a commercial nonlinear programming (NLP) solver [89] to incorporate higher order terms in our objective function. The nonlinear programming solver [89] uses the generalized reduced gradient (GRG) method to solve large-scale nonlinear problems [90].

We present experimental results on the approximation error in $\mathcal{P}_S^r$ when ILP is used to solve $\mathcal{P}_{TLS}$ versus when NLP is used. The relative defect-screening probability $\mathcal{P}_S^r$ was determined for a nonlinear objective function where the quadratic and cubic terms are considered in addition to the leading order term. Let $\mathcal{P}_{S-ILP}^r$ denote the relative defect-screening probability of the SoC obtained using a linear objective function (Equation (2.10)), and let $\mathcal{P}_{S-NLP}^r$ denote the relative defect-screening probability of the SoC using a nonlinear objective function. The nonlinear objective

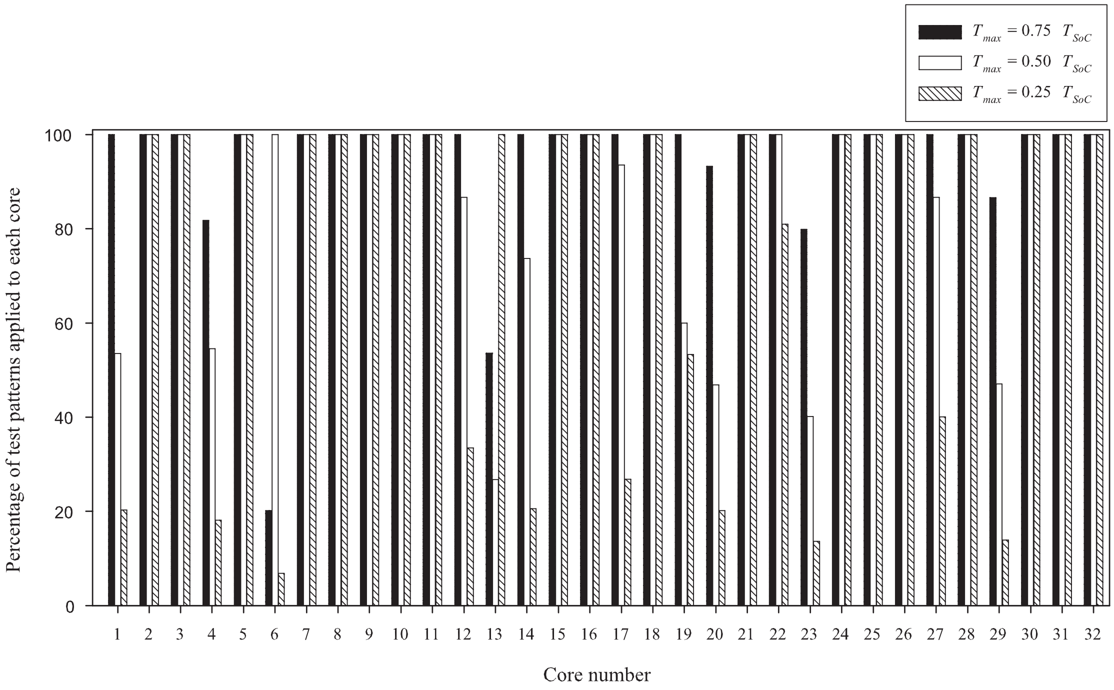


Figure 2.6: Percentage of test patterns applied to each core in p93791 for $W = 8$.

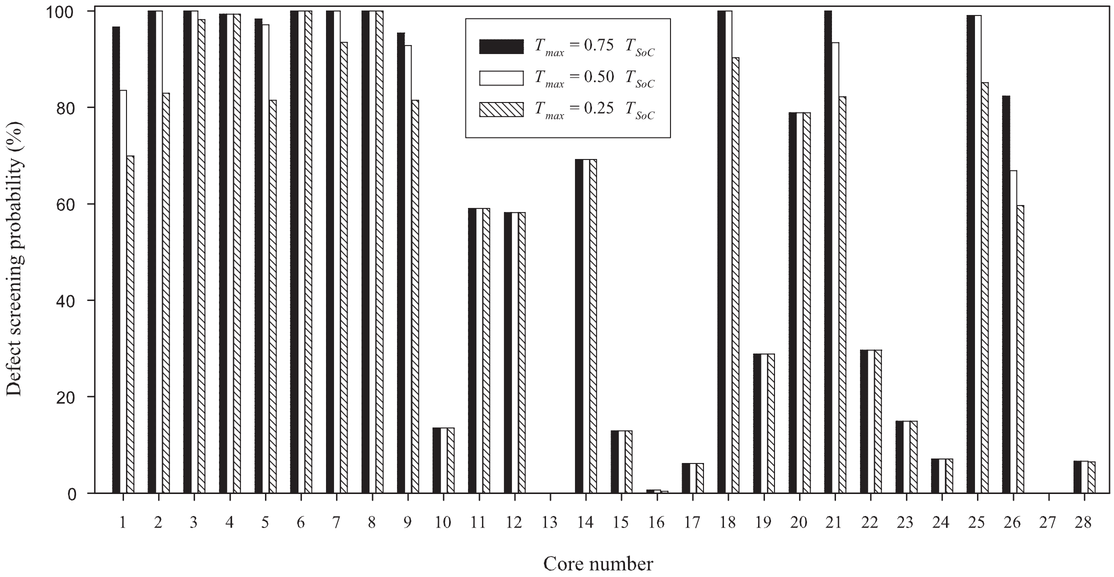


Figure 2.7: Relative defect-screening probabilities for the individual cores in p22810 for $W = 8$.

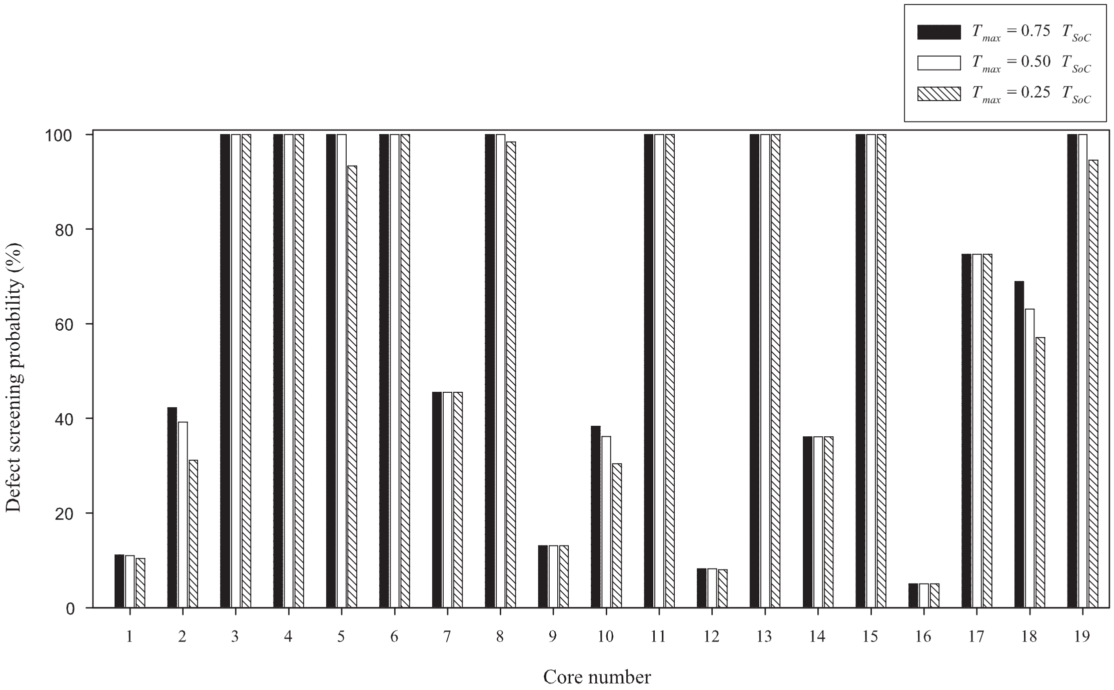


Figure 2.8: Relative defect-screening probabilities for the individual cores in p34392 for $W = 8$.

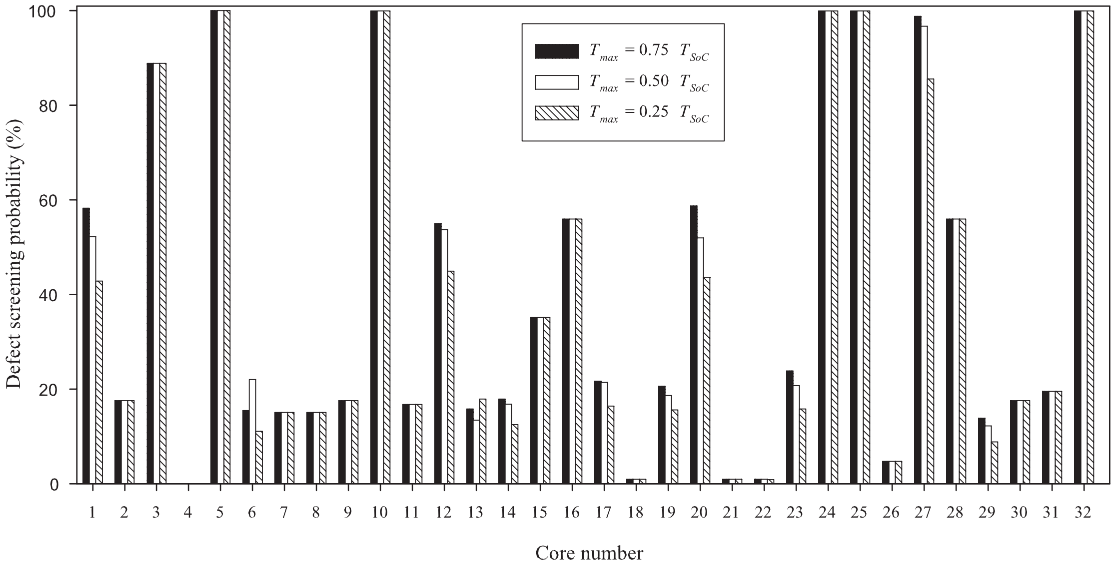


Figure 2.9: Relative defect-screening probabilities for the individual cores in p93791 for $W = 8$.

function that we use in our experiments is shown in Equation (2.11).

$$\text{Minimize } \mathcal{F} = \sum_{i=1}^{N} \left(\sum_{j=1}^{p_i} \delta_{ij}\epsilon_i(j) + \frac{(\delta_{ij}\epsilon_i(j))^2}{2} + \frac{(\delta_{ij}\epsilon_i(j))^3}{3} \right) \tag{2.11}$$

The relative magnitudes of the quadratic and cubic terms are negligible compared to the leading order term when the defect-escape probability of the core is negligible. We determine the approximation error as a measure to quantify the effect of these higher-order terms on $\mathcal{P}_S^r$. The approximation error is given by $\frac{\mathcal{P}_{S-ILP}^r - \mathcal{P}_{S-NLP}^r}{\mathcal{P}_{S-ILP}^r} \times 100\%$.

As in the case of any nonlinear optimization package, the commercial solver used [89] cannot guarantee finding a globally optimal solution in cases where there are distinct local optima and CPU time is limited. Knowledge of the convexity of the objective function and the constraints are essential to determine whether the nonlinear test-length selection problem will yield globally optimal solutions. In other words, if a function $f(x)$ has a second derivative in the interval $[a, b]$, a necessary and sufficient condition for it to be convex in that interval is that the second derivative $f''(x) \geq 0$, $\forall x$ in $[a, b]$ [91]. It is evident that the a second derivative exists for the objective function in Equation (2.11), and the function is convex; the solver therefore yields globally optimal solutions for the nonlinear test-length selection problem.

The approximation errors for the d695 SoC, and two "p" SoCs from Philips are shown in Table 2.3 respectively. The experimental results show that the relative defect-screening probabilities for the SoC are consistently higher when a linear objective function is used. The error in predicting the defect-screening probability, however, is less that 10% in most cases; our approximation is therefore reasonable for the benchmark circuits used in this work. The CPU time for *lpsolve*, to solve the ILP version of $\mathcal{P}_{TLS}$ for the largest SoC benchmark was less than a second. The time on the NLP solver [89] to solve $\mathcal{P}_{TLS}$ with the nonlinear objective function ranges from 4 minutes for the d695 SoC, to 26 minutes for the "p" SoCs from Philips. This clearly indicates that the nonlinear version of $\mathcal{P}_{TLS}$ is not scalable for large SoCs.

Table 2.3: Approximation error in $\mathcal{P}_S^r$ due to Taylor series approximation.

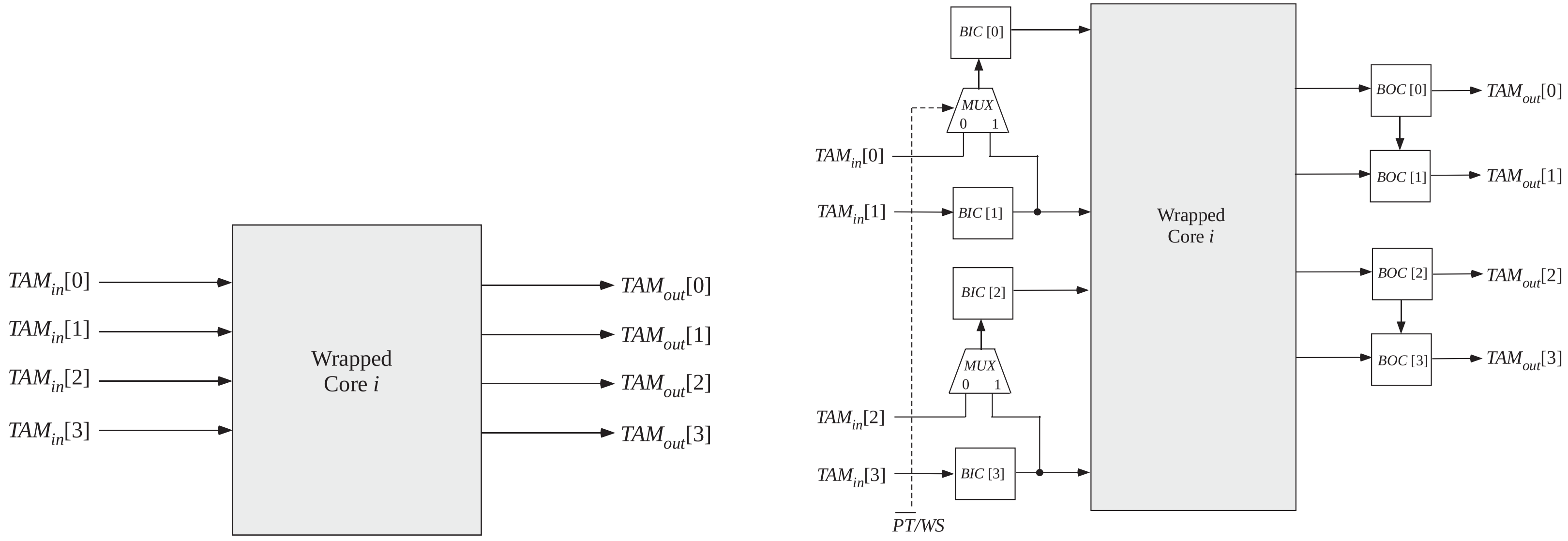
		Approximation Error (%)		
	W	$T_{max} = 0.75\ T_{SoC}$	$T_{max} = 0.5\ T_{SoC}$	$T_{max} = 0.25\ T_{SoC}$
d695	8	7.14	5.66	1.59
	16	7.55	5.25	4.28
	24	7.82	8.25	23.66
	32	11.19	8.62	9.80
p34392	8	1.31	7.35	7.40
	16	1.02	0.86	3.68
	24	0	0.86	3.87
	32	2.02	1.79	11.14
p22810	8	1.48	1.02	12.82
	16	1.48	0.53	18.43
	24	0.74	1.10	8.44
	32	2.15	36.08	36.18

2.4 Test data serialization

Suppose Core i is accessed from the SoC pins for package test using a TAM of width w_i (bits). Let us assume that for RPCT-based wafer sort, the TAM width for Core i is constrained to be w_i^* bits, where $w_i^* < w_i$. In order to access Core i using only w_i^* bits for wafer sort, the pre-designed TAM architecture for package test needs to be appropriately modified.

Figure 2.10(a) shows a wrapped core that is connected to a 4-bit wide TAM width ($w_i = 4$). For the same wrapped core, Figure 2.10(b) outlines a modified test access design that allows RPCT-based wafer-level test with $w_i^* = 2$. For wafer sort in this example, the lines $TAM_{out}[0]$, and $TAM_{out}[2]$ are not used. In order to ensure efficient test access architecture for wafer sort, serial-to-parallel conversion of the test data stream is necessary at the wrapper inputs of the core. A similar parallel-to-serial conversion is necessary at the wrapper outputs of the cores. Boundary input cells $BIC[0], \ldots, BIC[3]$, and boundary output cells $BOC[0], \ldots, BOC[3]$, which can operate in both a parallel load and a serial shift mode, are added at the I/Os of the wrapped core. Multiplexers are added on the input side of the core to enable the use of a smaller number of TAM lines for wafer sort. A global select signal $\overline{PT}/WS$ is used to choose either the package test mode ($\overline{PT}/WS = 0$) or the wafer sort mode ($\overline{PT}/WS = 1$). For the output side, the multiplexers are not needed; the test response can be serially shifted out to the TAM while the next pattern is serially shifted in to the boundary input cells. Note the above design is fully compliant with the IEEE 1500 standard [7] because no modifications are made to the standard wrapper cells.

We next explain how the test time for Core i is affected by the serialization process. Let $T_i(j)$ be the total testing time (in clock cycles) for core i if it is placed on TAM partition j of the SoC. Let $w_i(j)$ be the width of TAM partition j in the pre-designed TAM architecture. At the wafer level, if only $w_i^\star$ bits are available for TAM partition j, we assume, as in [92] for hierarchical SoC testing, that the w_i lines are distributed equally into $w_i^\star$ parts. Thus the wafer-level testing time for core i on TAM partition j equals $\lceil \frac{w_i(j)}{w_i^*(j)} \rceil \cdot T_i(j)$ clock cycles. In the example of Figure 2.10(b), the test time for core i due to serialization for is $T_i^*(j) = T_i(j) \cdot (4/2)$. Note that other TAM serialization methods can also be used for wafer sort. While TAM serialization can be integrated in an overall optimization problem, it is not considered here for the sake of simplicity.



(a) (b)

Figure 2.10: (a) Accessing a wrapped core for package test only (b) TAM design that allows RPCT-based wafer sort using a pre-designed wrapper/TAM architecture.

2.4.1 Test-length and TAM optimization problem: $\mathscr{P}_{TLTWS}$

Let us now consider an SoC with a top-level TAM width of W bits and suppose it has $\mathbf{B}$ TAM partitions of widths $w_1, w_2, \cdots, w_B$, respectively. For a given value of the maximum wafer level TAM width W^*, we need to determine appropriate TAM sub-partitions of widths w_1^*, w_2^*, $\ldots$, $w_{\mathbf{B}}^*$ such that $w_i^* \leq w_i$, $1 \leq i \leq \mathbf{B}$, and $w_1^* + w_2^* + \cdots + w_B^* = W^*$. The optimization problem $\mathscr{P}_{TLTWS}$ can now be formally stated as follows:

Problem $\mathscr{P}_{TLTWS}$: Given a pre-designed TAM architecture for a core-based SoC, the defect probabilities for each core in the SoC, maximum available test bandwidth at wafer sort W^* and the upper limit on the test time for the SoC at wafer sort T_{MAX}, determine (i) the total number of test patterns to be applied to each core, and (ii) the (reduced) TAM width for each partition, such that: (a) the overall testing time on each TAM partition does not exceed the upper bound T_{max}, and (b) the defect-screening probability $P(\bar{B}_i)$ for the SoC is maximized.

The objective function for the optimization problem is the same as that developed in Section 2.2.1 and is given by Equation (2.10). Due to serialization, the testing time for core i on TAM partition j, is given by $\lceil (w_i(j)/w_i^*(j) \rceil T_i(j)$ [92]. Therefore the test time of core i when it is tested with a reduced bitwidth of w_i^* is given by Equation (2.12).

$$T_i^* = \sum_{j=1}^{p_i} \delta_{ij} T_i(j) \left\lceil \frac{w_i(j)}{w_i^*(j)} \right\rceil \tag{2.12}$$

Let us now define a second binary indicator variable λ_{ik}, to ensure that every core in the SoC is tested using a single TAM width; this variable can be defined as follows:

$$\lambda_{ik} = \begin{cases} 1 & \text{if } w_i^* = 1/k \\ 0 & \text{otherwise} \end{cases}$$

It can be inferred from the above definition that $\sum_{k=1}^{w_i} \lambda_{ik} = 1$ and Equation (2.12) can now represented as $T_i^*(j) = \sum_{j=1}^{p_i} \sum_{k=1}^{w_i} \delta_{ij} T_i(j) \lambda_{jk} \lceil w_i \cdot k \rceil$. The nonlinear term in the constraint $\delta_{ij} \cdot \lambda_{ik}$ can be replaced with a new binary variable u_{ijk} by introducing two additional constraints:

$$\delta_{ij} + \lambda_{ik} \leq u_{ijk} + 1 \tag{2.13}$$

$$\delta_{ij} + \lambda_{ik} \geq 2 \cdot u_{ijk} \tag{2.14}$$

A constraint to ensure that every core in a TAM partition is tested with the same TAM width W_x^* is also necessary and can be represented as shown in Equation (2.15). The variable A_j denotes the set of cores that are assigned to TAM partition j. The constraint must be satisfied for every core in A_j.

$$\sum_{k=1}^{w_i} k \cdot \lambda_{ik} = W_x^* \tag{2.15}$$

The complete ILP model is shown in Figure 2.11. The number of variables and constraints in the ILP model determines the complexity of the problem. The number of variables in our ILP model is $\sum_{i=1}^{N}(p_i + w_i + p_i \cdot w_i)$, and the number of constraints is $2 \cdot N + 2\sum_{i=1}^{N}(p_i \cdot w_i) + B + 1$.

2.4.2 Experimental results: $\mathscr{P}_{TLTWS}$

We now present the experimental results for two SoCs from the ITC'02 SoC test benchmark suite [83]. We use the public domain ILP solver *lpsolve* for our experiments [88]. Since the objectives of our experiment are to select the number of test patterns in a time- and bitwidth-constrained wafer-sort environment, and at the same time maximize the defect-screening probability, we present the following results:

- Given values of W^* and T_{max} relative to T_{SoC}, the percentage of test patterns that must be applied for each individual core to maximize the defect-screening

Minimize $\mathcal{F} = \sum_{i=1}^{N} \left(\sum_{j=1}^{p_i} \delta_{ij}\epsilon_i(j) \right)$, subject to :

1) $\sum_{i=1}^{n_x} \left(\sum_{j=1}^{p_i} \sum_{k=1}^{w_i} \delta_{ij} T_i(j) \lambda_{jk} \lceil w_i \cdot k \rceil \right) \leq T_{max}; \forall x,\, 1 \leq x \leq B$
2) $\sum_{j=1}^{p_i} \delta_{ij} = 1; \forall i,\, 1 \leq i \leq N$
3) $\sum_{k=1}^{w_i} \lambda_{ik} = 1; \forall i,\, 1 \leq i \leq N$
4) $\sum_{k=1}^{w_i} k \cdot \lambda_{ik} = W_x^*; \forall Core_i \in A_j$
5) $\sum_{x=1}^{B} W_x^* \leq W^*$
6) $\delta_{ij} + \lambda_{ik} - 1 \leq u_{ijk}; \forall i, j, k$
7) $\delta_{ij} + \lambda_{ik} \geq 2 \cdot u_{ijk}; \forall i, j, k$

/* Constants : $\epsilon_i(j)$, T_{max} */
/* Variables : δ_{ij}, λ_{ik}, u_{ijk}; $1 \leq i \leq N$, $0 \leq j \leq p_i$ */

Figure 2.11: Integer linear programming model for $\mathscr{P}_{TLTWS}$.

probability for the SoC.

- The values of TAM partition widths $w_1^*, w_1^*, \cdots w_{\mathbf{B}}^*$ such that $w_1^* + w_1^* + \cdots + w_{\mathbf{B}}^* = W^*$.
- The relative defect-screening probability $\mathcal{P}_S^r$ for each core in an SoC, where $\mathcal{P}_S^r = \mathcal{P}_S / \mathcal{P}_S^{100}$ and $\mathcal{P}_S^{100}$ is the defect-screening probability if all 100% of the patterns are applied per core.
- The relative defect-screening probability for the SoC obtained using the ILP model.

We first present results on the number of patterns determined for the cores. The results for the d695 benchmark SoC is presented in Figure 2.12 for three values of T_{max}: T_{SoC}, $0.75T_{SoC}$ and $0.5T_{SoC}$. The fraction of test patterns applied per core is found to be different in each case to maximize the defect-screening probability.

Table 2.4: Relative Defect-Screening Probabilities Obtained Using $\mathscr{P}_{TLTWS}$ ($W = 32$).

SoC	W^*	$T_{max} = T_{SoC}$		$T_{max} = 0.75T_{SoC}$		$T_{max} = 0.5T_{SoC}$	
		Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability	Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability	Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability
d695	8	(5,1,2)	0.3982	(5,1,2)	0.2907	(4,1,3)	0.1058
	12	(5,1,6)	0.4426	(5,1,6)	0.3272	(5,3,4)	0.2631
	16	(10,3,3)	0.9064	(10,3,3)	0.7279	(10,3,3)	0.4306
a586710	8	(1,4,3)	0.7294	(1,4,3)	0.6142	(1,4,3)	0.4623
	12	(1,7,4)	0.7519	(1,7,4)	0.6682	(1,7,4)	0.5191
	16	(1,8,7)	0.7621	(1,8,7)	0.6682	(1,8,7)	0.5191

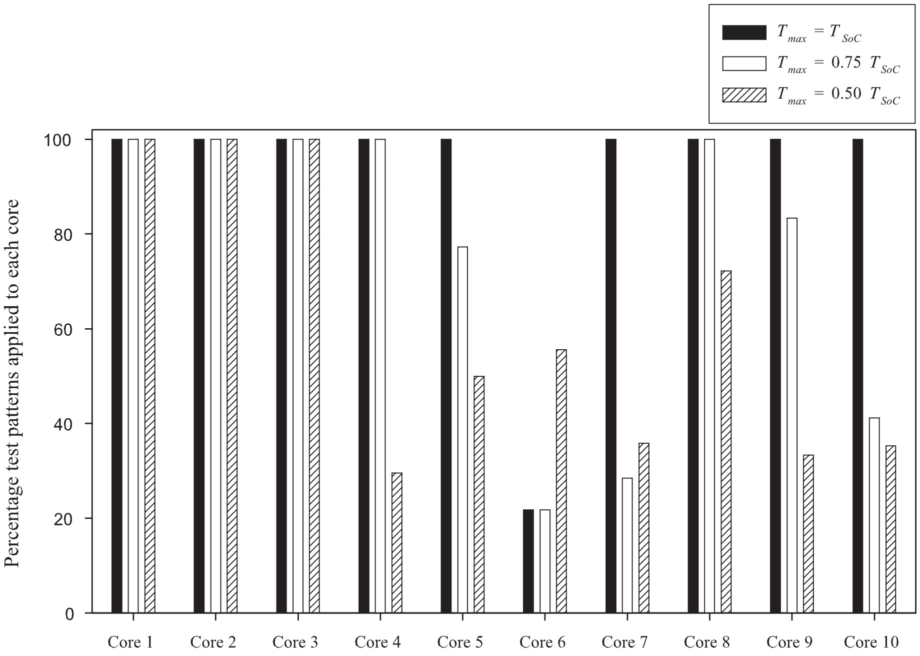


Figure 2.12: Percentage of test patterns applied to each core in d695 when $W^* = 16$ and $W = 32$.

Results are reported only for $W^* = 16$ and $W = 32$; similar plots are obtained for different values of W^* and W. Figure 2.13 illustrates the defect-screening probabilities for the cores in the d695 benchmark for the above-mentioned test case.

We summarize the results for two benchmark SoCs in Table 2.4.2 for three different values of W^* and $W = 32$. The relative defect-screening probabilities $\mathcal{P}_S$ and TAM partition widths to be used at wafer sort, obtained using $\mathscr{P}_{TLTWS}$, are enumerated for both benchmark SoCs. The ILP-based technique takes up to 3 hours of CPU time on a 2.4 GHz AMD Opteron processor with 4 GB of memory for d695, when $W^* = 16$ and $W = 32$. The results show that a significant portion of the faulty dies can be screened at wafer sort using the proposed technique.

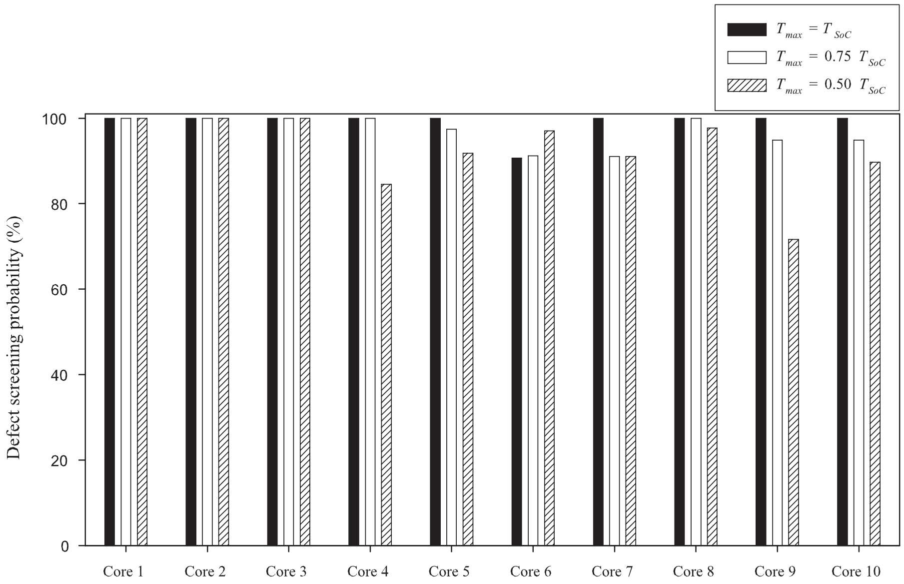


Figure 2.13: Relative defect-screening probabilities for the individual cores in d695 when $W^* = 16$ and $W = 32$.

2.4.3 Enumeration-based TAM width and test-length selection

The ILP-based approach in Section 2.2 is efficient only for small SoCs. However, due to its large size, it may not scale well for SoCs with a large number of cores. It is therefore necessary to develop an alternative technique that can handle larger SoC designs. We next propose an efficient heuristic approach $\mathscr{P}_{e-TLTWS}$ based on a combination TAM partition-width enumeration and ILP.

Our enumeration approach is based on the "odometer" principle used in a car odometer. Each digit of a car odometer here corresponds to a TAM partition width at wafer sort. Each digit can take values between 1 and the upper limit fixed by the TAM architecture designed for package test. We first increase the least significant digit if possible, and next roll the digit over to one and increase the next least-significant digit. The implementation of the enumeration approach for determining the optimal TAM partition widths and test-lengths can be done using the following sequence of procedures:

(i) Given the number of TAM partitions **B** and an upper limit on the maximum TAM width W^*, we first enumerate all possible TAM partition combinations. This enumeration can be done following the principle of a **B**-bit odometer, where each bit corresponds to the width of each TAM partition. The odometer resets to one as opposed to zero in the case of a conventional odometer (the maximum value that the i^{th} bit can take before a reset is w_i). At every increment in the odometer, we check whether $\sum_{i=1}^{B} w_i^* = W^*$.

All possible TAM partitions that meet the above condition are recorded as a valid partition. We illustrate the above enumeration procedure with a small example. Let us consider an SoC whose TAM architecture is fixed and designed for 5 bits, and partitioned into three TAM partitions of widths 2, 3, and 1 respectively. The possible TAM enumerations for the above partitions are $\{\langle 1,1,1\rangle, \langle 1,2,1\rangle, \langle 1,3,1\rangle, \langle 2,1,1\rangle, \langle 2,2,1\rangle, \langle 2,3,1\rangle\}$. If we consider W^* to be 4, then the valid TAM partitions are $\{\langle 1,2,1\rangle, \langle 2,1,1\rangle\}$.

(ii) For each valid TAM partition calculated in Step (i), we apply the test-length selection procedure $\mathscr{P}_{TLS}$. We calculate the defect-screening probability for the SoC from the results obtained using $\mathscr{P}_{TLS}$.

(iii) If the defect-screening probability of the new partition is greater than the previous partition, we store it as the new defect-screening probability, and store this partition as the current optimal partition.

(iv) We repeat this procedure until all possible TAM partitions are enumerated.

Experimental results obtained using the $\mathscr{P}_{e-TLTWS}$ procedure are summarized in Table 2.5. The results are represented in a similar fashion as in Table 2.4. The values of the defect screening probabilities $\mathcal{P}_S$ for five benchmark circuits [83], as well as the recommended TAM partition widths for wafer-sort are shown in the table. The number of patterns determined using $\mathscr{P}_{e-TLTWS}$ for the p34392 SoC is illustrated in

Fig. 2.14. The results are shown for three values of T_{max}: T_{SoC}, $0.75T_{SoC}$ and $0.5T_{SoC}$. Results are reported only for $W^* = 16$ and $W = 32$; similar plots are obtained for a range of values of W^* and W. Fig. 2.15 illustrates the relative defect-screening probabilities for the cores in the p34392 benchmark for the above-mentioned test case. The heuristic method results in lower defect-screening probability for most cases compared with the ILP-based method; for higher values of W^*, the difference in defect screening probability between the two methods decreases. The computation time for the largest benchmark SoC p93791 was only 4 minutes, hence this approach is suitable for large designs.

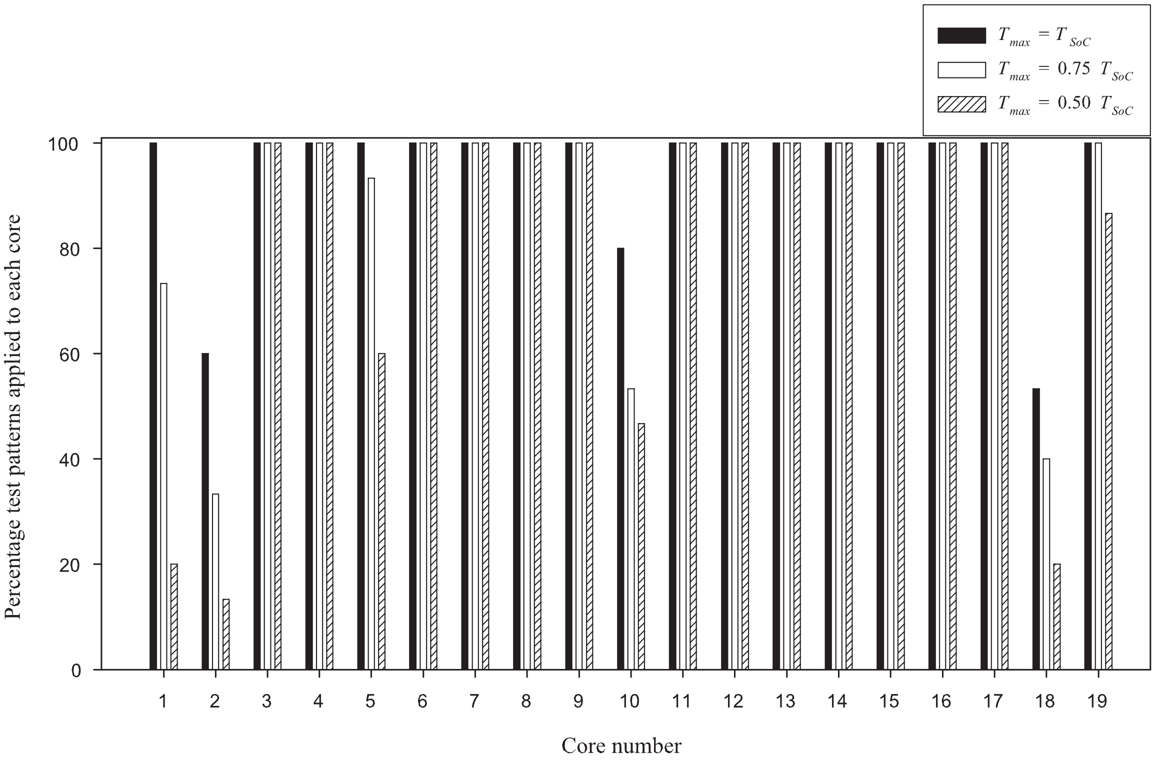


Figure 2.14: Percentage of test patterns applied to each core in p34392 when when $W^* = 16$ and $W = 32$.

Table 2.5: Relative Defect-Screening Probabilities Obtained Using $\mathscr{P}_{e-TLTWS}$.

SoC	W^*	$T_{max} = T_{SoC}$		$T_{max} = 0.75T_{SoC}$		$T_{max} = 0.5T_{SoC}$	
		Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability	Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability	Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability
a586710	8	(1,5,2)	0.5732	(1,5,2)	0.5341	(1,5,2)	0.3319
	12	(2,6,4)	0.7014	(2,6,4)	0.5789	(2,6,4)	0.4449
	16	(2,9,5)	0.7118	(2,9,5)	0.5837	(2,9,5)	0.4580
d695	8	(5,1,2)	0.5392	(4,2,2)	0.5471	(5,1,2)	0.3102
	12	(7,2,3)	0.8139	(7,3,2)	0.5542	(7,2,3)	0.4116
	16	(9,2,5)	0.8543	(9,3,4)	0.7022	(8,2,6)	0.5231
p34392	8	(3,3,2)	0.3385	(3,2,3)	0.2275	(3,2,3)	0.1110
	12	(4,5,3)	0.6382	(4,4,4)	0.4360	(4,4,4)	0.2180
	16	(6,7,3)	0.8010	(5,7,4)	0.5968	(4,7,5)	0.2948
p22810	8	(3,3,2)	0.1331	(3,3,2)	0.0580	(3,3,2)	0.0098
	12	(4,6,2)	0.1891	(4,5,3)	0.1800	(3,6,3)	0.0333
	16	(5,6,5)	0.6186	(6,6,4)	0.3841	(6,6,4)	0.1495
p93791	8	(2,4,2)	0.0606	(2,4,2)	0.0165	(2,4,2)	0.0050
	12	(3,6,3)	0.2228	(3,7,2)	0.0949	(3,7,2)	0.0189
	16	(4,8,4)	0.5018	(4,8,4)	0.2201	(4,8,4)	0.0615

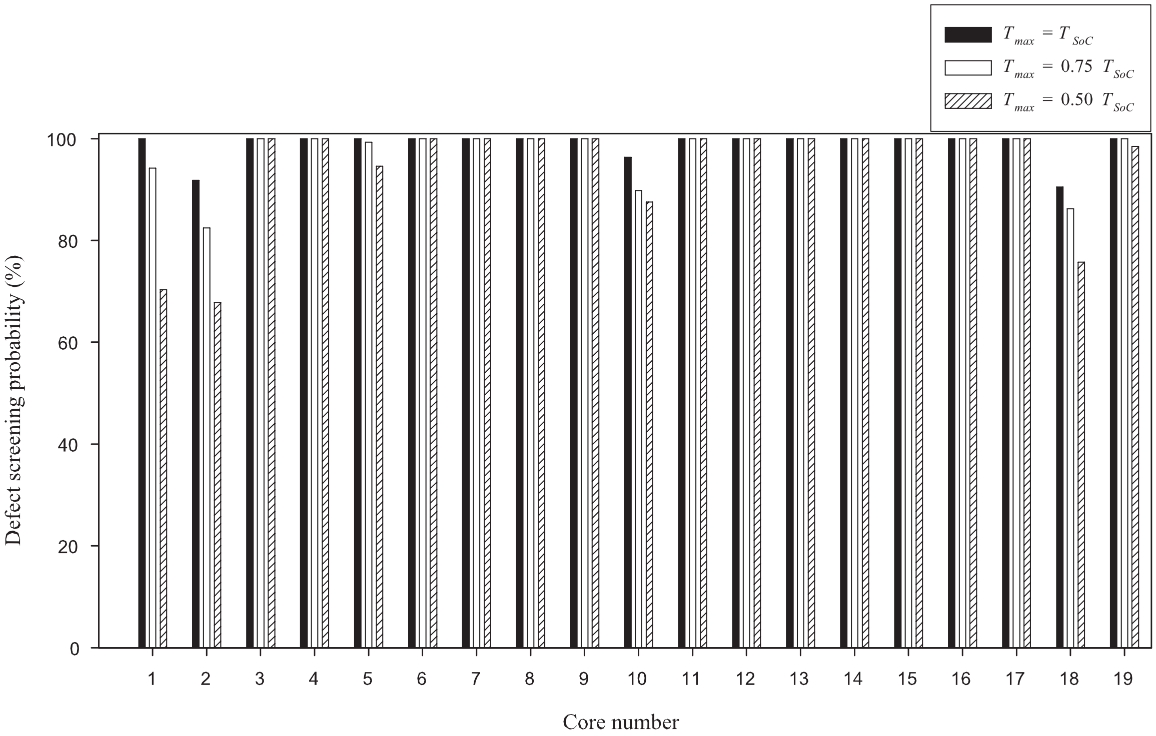


Figure 2.15: Relative defect-screening probabilities for the individual cores in p34392 when $W^* = 16$ and $W = 32$.

2.4.4 TAM width and test-length selection based on geometric programming

Geometric programming (GP) problems are convex optimization problems that are similar to linear programming problems [93]. A GP is a mathematical problem of the form

$$\begin{aligned} \text{Minimize} \quad & f_0(x) \\ \text{subject to} \; & f_i(x) \le 1, \; i = 1, \cdots, m \\ & g_i(x) = 1, \; i = 1, \cdots, p \end{aligned}$$

where f_i are posynomial functions, g_i are monomials, and x_i are the optimization variables; it is implicitly assumed that the optimization variables are positive, i.e., $x_i > 0$ [93]. Mixed-integer GPs are a class of problems that are hard to solve [93]. The problem $\mathscr{P}_{TLTWS}$ can be modeled as a mixed-integer GP (MIGP) problem. In this chapter, we employ a heuristic method to solve the MIGP problem $\mathscr{P}_{gp-TLTWS}$ for test-length and TAM width selection. Using heuristic methods, approximate solutions can be found in a reasonable amount of time; however, the optimality of the solution cannot be guaranteed. Before we describe the GP-based heuristic method, we need to modify the objective function to make it amenable for further analysis. The objective of $\mathscr{P}_{TLTWS}$ is to maximize the defect-screening probability, $\mathcal{P}_S = 1 - \prod_{i=1}^{N} \mathcal{P}(B_i)$. This is equivalent to the following minimization-based objective function.

$$\text{Minimize } \mathcal{G} = \prod_{i=1}^{N} \left(\sum_{j=1}^{p_i} \delta_{ij} \epsilon_i(j) \right)$$

Minimize $\mathcal{G} = \prod_{i=1}^{N} \left(\sum_{j=1}^{p_i} \delta_{ij}\epsilon_i(j) \right)$, subject to :

1) $\frac{\sum_{i=1}^{n_x} \left(\sum_{j=1}^{p_i} \sum_{k=1}^{w_i} \delta_{ij} T_i(j) \lambda_{jk} \lceil w_i \cdot k \rceil \right)}{T_{max}} \leq 1; \forall x,\ 1 \leq x \leq B$
2) $\sum_{j=1}^{p_i} \delta_{ij} = 1; \forall i,\ 1 \leq i \leq N$
3) $\sum_{k=1}^{w_i} \lambda_{ik} = 1; \forall i,\ 1 \leq i \leq N$
4) $\frac{\sum_{k=1}^{w_i} k \cdot \lambda_{ik}}{W_x^*} = 1; \forall Core_i \in A_j$
5) $\frac{\sum_{x=1}^{B} W_x^*}{W^*} = 1$

/* Constants : $\epsilon_i(j)$, T_{max}, W^* */

/* Variables : δ_{ij}, λ_{ik}; $1 \leq i \leq N$, $0 \leq j \leq p_i$ */

Figure 2.16: Geometric programming model for $\mathscr{P}_{TLTWS}$.

The constraints for the optimization problem described in Section 2.4.1 can be easily modified for use in the MIGP problem. The complete MIGP problem for $\mathscr{P}_{TLTWS}$ is shown in Figure 2.16. We use GP relaxation to transform the MIGP problem to a general GP problem that can be solved using commercial tools [94]. To obtain an approximate solution of the MIGP problem, the MIGP is relaxed to a GP and solved using [94]; the result obtained in this way is an upper bound on the optimal value of the objective function for the MIGP. The values of the variables obtained after relaxation are then simply rounded towards the nearest integer. The heuristic then iteratively reassigns the values of the variables such that the constraints are satisfied while maximizing the defect-screening probability for the SoC. The heuristic used to solve $\mathscr{P}_{gp-TLTWS}$ consists of the following steps:

1. In the first step of this procedure, we relax the MIGP $\mathscr{P}_{TLTWS}$ to a GP problem. The relaxation essentially means that the binary indicator variables used in the optimization problem can take non-integer values.

2. We then use [94] to solve the relaxed MIGP problem. The resulting values of the indicator variables δ_{ij} are sorted for each core i. The highest value of δ_{ij} for each core is rounded to unity, while the remaining variables are rounded down to zero.

3. The procedure then assigns the smallest value of TAM width to each core in the SoC; i.e., $\lambda_{i1} = 1, \forall i$. For the smallest value of TAM widths assigned to the cores, the test time for each TAM partition is calculated.

4. The procedure then iteratively assigns additional TAM width to the TAM partition with the maximum test time. This is repeated until $\sum_{i=1}^{B} w_i^* = W^*$.

5. Once the TAM widths for RPCT are determined, we check to determine if $\sum_{Core_i \in A_j} T_i^* \leq T_{max}, 1 \leq j \leq B$. If violations in test time constraints are observed, we identify a core in each TAM partition for which a reduction in the number of applied patterns results in a minimal decrease in the overall defect-screening probability. For each TAM partition $j, 1 \leq j \leq B$, where a violation in test time is observed, we chose a particular $Core_i \in A_j$ such that a decrease in the number of applied patterns Δp_i^* results in a minimal decrease in $\epsilon_i(p_i^*)$; we consider different values of Δp_i^* in the range $1 \leq \Delta p_i^* \leq 5$ in our

experiments, and choose the value that results in maximum defect screening $\mathcal{P}_S^r$ for the SoC. This procedure, searches for a core in each TAM partition, which yields a maximum value for $\theta_i \cdot (fc_i(p_i^*) - (fc_i(p_i^* - \Delta p_i^*)$. This is repeated until the time constraint on all TAM partitions are satisfied.

6. The relative defect-screening probability $\mathcal{P}_S^r = \mathcal{P}_S/\mathcal{P}_S^{100}$ for each core in the SoC is then calculated; $\mathcal{P}_S^{100}$ is the defect-screening probability if all 100% of the patterns are applied per core. This information is used to determine the relative defect-screening probability for the SoC.

Experimental results obtained using the GP-based heuristic procedure are summarized in Table 2.6. The results are represented in a similar fashion as in Table II. The relative defect-screening probability obtained using the GP-based heuristic is greater than that obtained using the enumerative heuristic technique and less than that obtained using the ILP method. The computation time ranges from 6 minutes for the a586710 SoC, to 51 minutes for the p93791 SoC.

2.4.5 Approximation error in $\mathcal{P}_S^r$

We present experimental results on the approximation error in $\mathcal{P}_S^r$ when ILP and heuristic methods are used to solve $\mathcal{P}_{TLTWS}$ versus when NLP and GP-based methods are used. We use a commercial solver [94] for the GP-based heuristic method. The relative defect-screening probability was determined for a nonlinear objective (Equation (2.11)) function using [89], where the quadratic and cubic terms are considered in addition to the leading order term; this procedure is similar to the procedure described in Section 2.3.

Let $\mathcal{P}_{S-ILP}^r$ denote the relative defect-screening probability of the SoC obtained using a linear objective function, $\mathcal{P}_{S-e-TLTWS}^r$ the defect-screening probability using the enumerative heuristic, $\mathcal{P}_{S-NLP}^r$ denote the relative defect-screening prob-

Table 2.6: Relative Defect Screening Probabilities Obtained Using the GP-based Heuristic Method.

		$T_{max} = T_{SoC}$		$T_{max} = 0.75T_{SoC}$		$T_{max} = 0.5T_{SoC}$	
SoC	W^*	Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability	Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability	Optimal Distribution (w_1, w_2, w_3)	Defect-Screening Probability
a586710	8	(4,2,2)	0.7226	(4,1,3)	0.5833	(4,1,3)	0.4594
	12	(6,3,3)	0.7446	(6,3,3)	0.6391	(6,3,3)	0.5138
	16	(9,3,4)	0.7582	(8,3,5)	0.6435	(8,3,5)	0.5120
d695	8	(4,1,3)	0.5027	(4,1,3)	0.5288	(4,1,3)	0.3014
	12	(6,1,5)	0.7962	(6,2,4)	0.5532	(6,2,4)	0.3961
	16	(8,2,6)	0.8420	(8,2,6)	0.6931	(8,2,6)	0.5090
p34392	8	(3,4,1)	0.3440	(3,3,2)	0.2330	(3,3,2)	0.1150
	12	(3,4,4)	0.6473	(3,4,4)	0.4455	(3,4,4)	0.2251
	16	(6,6,4)	0.8072	(6,5,5)	0.6081	(6,5,5)	0.3002
p22810	8	(3,2,3)	0.1346	(3,1,4)	0.0598	(3,1,4)	0.0100
	12	(4,5,3)	0.1911	(4,6,2)	0.1848	(4,6,2)	0.0322
	16	(4,5,7)	0.6246	(5,4,7)	0.3892	(5,4,7)	0.1508
p93791	8	(4,3,1)	0.0620	(4,3,1)	0.0170	(4,2,2)	0.0051
	12	(2,6,4)	0.2285	(2,6,4)	0.0977	(2,6,4)	0.0193
	16	(3,7,6)	0.5119	(3,8,5)	0.2216	(3,8,5)	0.0619

ability of the SoC using a nonlinear objective function, and $\mathcal{P}^r_{S-GP}$ the relative defect-screening probability using the GP-based heuristic method. We determine the approximation error as a measure to quantify the effect of these higher-order terms on $\mathcal{P}^r_S$. The approximation error obtained using the ILP method is determined as $\delta_{ILP} = \frac{\mathcal{P}^r_{SILP} - \mathcal{P}^r_{SNLP}}{\mathcal{P}^r_{SNLP}} \times 100\%$. The approximation errors obtained using the heuristic and GP are similarly determined as $\delta_{Heur} = \frac{\mathcal{P}^r_{SHeur} - \mathcal{P}^r_{SNLP}}{\mathcal{P}^r_{SNLP}} \times 100\%$ and $\delta_{GP} = \frac{\mathcal{P}^r_{SGP} - \mathcal{P}^r_{SNLP}}{\mathcal{P}^r_{SNLP}} \times 100\%$ respectively.

As it is evident from the above equations, the results obtained using the nonlinear programming solver is used as a baseline case. This is because the results obtained using GP-based heuristic are only bounds (upper bounds on the relative defect-screening probability), and the results obtained using ILP and the enumerative heuristic method are not optimal. The "p" benchmarks do not consider solutions obtained using ILP because of the lack of a suitable solver to solve problems of this size. The approximation errors for the benchmark circuits are presented in Tables 2.6-2.7. The time needed by the NLP solver [89] to solve $\mathcal{P}_{TLTWS}$ with the nonlinear objective function ranges from 6 minutes for the d695 SoC, to 4 hours for the "p" SoCs from Philips. This clearly indicates that the nonlinear version of $\mathcal{P}_{TLTWS}$ is not scalable for large SoCs. The time to solve the GP-based heuristic ranges from 2 minutes for the d695 SoC, to 45 minutes for the "p" SoCs. The GP-based heuristic can therefore be used to quickly determine bounds on $\mathcal{P}^r_S$.

2.5 Summary

We have formulated a test-length selection problem for wafer-level testing of core-based SoCs. This is the first attempt to formulate a test-length selection problem for wafer sort of core-based SoCs. To solve this problem, we first showed how defect

Table 2.7: Approximation error in relative defect-screening probability for d695 and a586710.

SoC	W^*	$T_{max} = T_{SoC}$			$T_{max} = 0.75T_{SoC}$			$T_{max} = 0.5T_{SoC}$		
		δ_{Heur}	δ_{GP}	δ_{ILP}	δ_{Heur}	δ_{GP}	δ_{ILP}	δ_{Heur}	δ_{GP}	δ_{ILP}
d695	8	4.36	1.59	0.17	7.77	4.82	0.37	15.09	11.81	3.31
	12	3.83	1.58	0.15	7.48	7.48	0.58	12.73	10.45	2.94
	16	3.23	1.74	0.25	7.10	5.72	1.04	11.31	9.61	2.36
a586710	8	2.47	1.52	0.12	3.77	1.46	0.54	5.40	4.76	0.94
	12	2.18	1.46	0.32	3.27	1.22	0.41	4.87	3.79	0.79
	16	2.05	1.53	0.28	3.03	0.78	0.46	4.36	2.93	1.03

Table 2.8: Approximation error in relative defect-screening probability for the "p" SoCs.

SoC	W^*	$T_{max} = T_{SoC}$		$T_{max} = 0.75T_{SoC}$		$T_{max} = 0.25T_{SoC}$	
		δ_{Heur}	δ_{GP}	δ_{Heur}	δ_{GP}	δ_{Heur}	δ_{GP}
p22810	8	2.91	4.09	4.36	7.68	5.40	7.18
	12	1.82	2.92	4.29	7.09	5.31	8.19
	16	1.28	2.26	3.06	4.43	3.62	4.54
p34392	8	0.97	2.62	3.48	5.99	6.54	10.33
	12	1.12	2.56	3.25	5.49	7.55	11.07
	16	1.85	2.64	2.84	4.78	5.67	7.62
p93791	8	4.30	6.62	5.75	8.97	6.48	8.92
	12	6.16	8.87	7.10	10.28	8.21	10.72
	16	4.90	7.01	6.53	7.23	8.35	9.15

probabilities for the individual cores in an SoC can be obtained using statistical modeling techniques. The defect probabilities were then used in an ILP model to solve the test-length selection problem. The ILP approach takes less than a second for the largest SoC test benchmarks from Philips. Experimental results for the ITC'02 SoC test benchmarks show that the ILP-based method can contribute significantly to defect-screening at wafer sort. A heuristic method that scales well for larger SoCs has also been presented.

We have also formulated a test-length and a TAM width selection problem for wafer-level testing of core-based digital SoCs. To the best of our knowledge, this is the first attempt to incorporate TAM-width-selection in the wafer-level SoC test flow. Experimental results for the ITC'02 SoC test benchmarks using the optimal method and the enumeration based approach show that the proposed approach can contribute to effective defect screening at wafer sort.

Chapter 3 presents a wafer-level defect screening technique for core-based mixed-signal SoCs. A cost model is also formulated to study the impact of the defect-screening technique on overall cost savings.

Chapter 3

Defect Screening for "Big-D/Small-A" Mixed-Signal SoCs

Conventional test techniques for mixed-signal circuits requires the use of a dedicated analog test bus and an expensive mixed-signal ATE [95, 96]. In this chapter, we present a correlation-based signature analysis technique for mixed-signal cores in a SoC [97]. This method is specifically developed for defect screening at the wafer level using low-cost digital testers, and with minimal dependence on mixed-signal testers.

A comprehensive cost model is needed to evaluate the effectiveness of wafer-level testing, and its impact on test and packaging cost. We develop a cost model and use it to quantify the benefits derived from wafer-level testing of both analog and digital cores. Correction factors, which account for the misclassification of dies under test, are incorporated in the cost model. Experimental results involving the wafer-level test technique as well as the cost model are presented for an industrial mixed-signal SoC. The results show that a significant reduction in product cost can be obtained using wafer-level testing and the proposed signature analysis method.

The remainder of the chapter is organized as follows. Section 3.1 describes the proposed signature analysis method for wafer-level test of analog cores. Simulation results are presented to evaluate the signature analysis method. Section 3.2 describes the cost model for a generic mixed-signal SoC. Section 3.3 details the reduction in product cost that can be obtained using wafer-level testing for an industrial mixed-signal SoC. Finally, Section 3.4 concludes this work.

3.1 Wafer-level defect screening: Mixed-signal cores

Test procedures for data converters can be classified as being based on either spectral-based tests or code density tests. Spectral-based test methods [96] usually involve the use of a suitable transform, such as the Fourier Transform, to analyze the output. These methods are used to determine the dynamic test parameters of the data converter. On the other hand, code density tests are based on the constructions of histograms of the individual code counts [98]. The code counts of the data converter-under-test are then analyzed and compared with the expected code counts to determine its static parameters. Recent work in mixed-signal testing has focused on spectral-based frequency domain tests, due to the inherent advantage of test time over the code density tests. In [96], a test flow process is described, that uses only the dynamic tests. A case study on sample data converters presented in [96] claims that 96% of faults involving both static and dynamic specifications can be detected without using the code density test technique. It is important to note that the procedure described in [96] is aimed at production testing. In [99], it has been shown that frequency-domain-based signature analysis helps in suppressing non-idealities associated with the test data, and it serves as a robust mechanism for enhancing fault coverage and reducing false alarms.

In effect, a mixed-signal path can be sandwiched between a pair of complementary data converters to generate a mixed-signal core driven by digital inputs and outputs [53]. Testing this mixed-signal path, which is a basic building block in most "big-D/small-A" SoC designs, holds key to cost effective testing using low cost digital testers. The inadequacy of analog tests and their lack of effectiveness at wafer sort to accurately measure test parameters and identify faulty dies have been highlighted in [33] and [56]. A new defect screening technique for mixed-signal cores at wafer sort is needed for the following reasons:

- Time-domain signature analysis techniques have extremely low tolerance to noise, since the measured signature can be incorrect even for single bit errors [100].
- Noisy signals and imprecise test clocks at wafer sort lead to distortion in the value of the dynamic parameters of such a signal to noise ratio (SNR), which directly affects the effective number of bits for the data converter. The lower-order bits of the data converter, in the presence of noise, convert noise rather than the signal itself. In such circumstances, the comparison of the data converter with a pre-specified signature, inevitably leads to increased yield loss.
- Test signals which are more linear than the linearity of the device under test (DUT), are prescribed as a requirement for successful testing of data converters [101]. This cannot be guaranteed in "big-D/small-A" SoC designs, as the digital-to-analog converters (DACs) are used to provide test stimuli to the analog-to-digital converters (ADCs), when configured in a loop-back mode.

Measurement inaccuracies associated with a mixed-signal test and measurement environment are described in [53, 30]. These problems can lead to a degradation in the quality of the measurements made; these effects are more pronounced at wafer sort [30, 56]. As a result, yield loss and test escape are more likely at the wafer-level.

Test procedures examine the output response of the circuit and compare it to a pre-determined "acceptable" signature. In light of all the possible error sources during wafer sort, a reliable acceptable signature is hard to derive because it requires the modeling of all possible errors. To address the above problems, outlier analysis has been extensively used in the IDDQ testing of digital circuits [57, 58]. We employ a similar pass/fail criterion in the proposed wafer-level testing approach. To perform such an analysis, we first require a measurable parameter for each core. In IDDQ testing, this data comes in the form of supply current information. However, in spectral analysis, the information obtained as a signature is spread over multiple data points, where each data point represents the power associated with the corresponding frequency bin. It is therefore necessary to encode this information as a single parameter corresponding to each individual core. We propose two correlation-based test methods to achieve this goal. These methods are referred to as the *mean-signature*- and *golden-signature*-based correlation techniques.

3.1.1 Signature analysis: Mean-signature-based-correlation (MSBC)

In [99], the authors use the correlation between a reference spectrum and the spectrum of the circuit under test as a pass/fail criterion. The reference spectrum serves as an acceptable signature, and is used for comparison with the spectrum of the circuit under test. Such a reference signature is called an Eigen signature [99]. The sensitivities to changes in the shape of the spectrum of the device-under-test from the Eigen signature can be quantified by means of a correlation parameter. The correlation is a fraction that lies between 0 and 1, and it serves as a single measurable parameter for each individual die.

The characteristic spectrum X_i of the i^{th} core-under-test in a batch of m identical cores is obtained using a P-point Fast Fourier Transform (FFT) and is defined as: $X_i = \{x_{i1}, x_{i2}, \cdots x_{iP}\}$, $1 \leq i \leq m$. The elements $x_{i1}, x_{i2}, \cdots x_{iP}$ in the above spectrum denote the power associated with the corresponding frequency bin. Ideally, the spectrum of each die should be correlated to a set of averages of the spectra of m dies tested under similar ambient operating conditions. The Eigen signature E is determined as the set of averages of the spectra of m identical cores-under-test and can be defined as: $E = \{(\sum_{i=1}^{m} x_{i1})/m, (\sum_{i=1}^{m} x_{i2})/m, \cdots, (\sum_{i=1}^{m} x_{iP})/m\}$. In particular, if the number of good dies is appreciably larger than the number of defective ones, the Eigen signature contains the information needed to classify the good dies from the defective ones. Since both X_i and E are random variables, let $\overline{X}_i$ and $\overline{E}$ represent the mean of X_i and E respectively. The correlation between the Eigen spectrum and that of the circuit under test can now be defined using Equation (3.1) as:

$$\mathrm{corr}(X_i, E) = \frac{\sum_{j=1}^{P}(x_{ij} - \overline{X}_i)\left(\frac{\sum_{i=1}^{m} x_{ij}}{m} - \overline{E}\right)}{\left[\sum_{j=1}^{P}(x_{ij} - \overline{X}_i)^2 \sum_{j=1}^{P}\left(\frac{\sum_{i=1}^{m} x_{ij}}{m} - \overline{E}\right)^2\right]^{1/2}} \tag{3.1}$$

3.1.2 Signature analysis: Golden-signature-based-correlation (GSBC)

For the MSBC technique, the collection of spectral signatures requires the storage of spectral information of a number of dies before a pass/fail decision can be made. While this information does not have to reside in the main memory of the tester, storing and handling such a large amount of data may be inconvenient. It may be desirable to use a pre-defined *golden-signature* for correlation during wafer sort. It is important to note that the use of a pre-defined spectrum as the *golden signature* does not hamper outlier analysis. The *golden-signature* spectrum is obtained *a priori*, by assuming ideal and fault-free operating conditions for the circuit under test. The correlation parameter can still be used to identify the possible faulty dies. The correlation parameters are estimated in the same way as in Section 3.1.1. The only difference here lies in the use of a *golden signature* as the Eigen signature. The test flow for both methods is described in Figure 3.1.

The next step in signature analysis is to set a threshold to determine the pass/fail criterion for each die. As explained previously, due to all the non-idealities in the

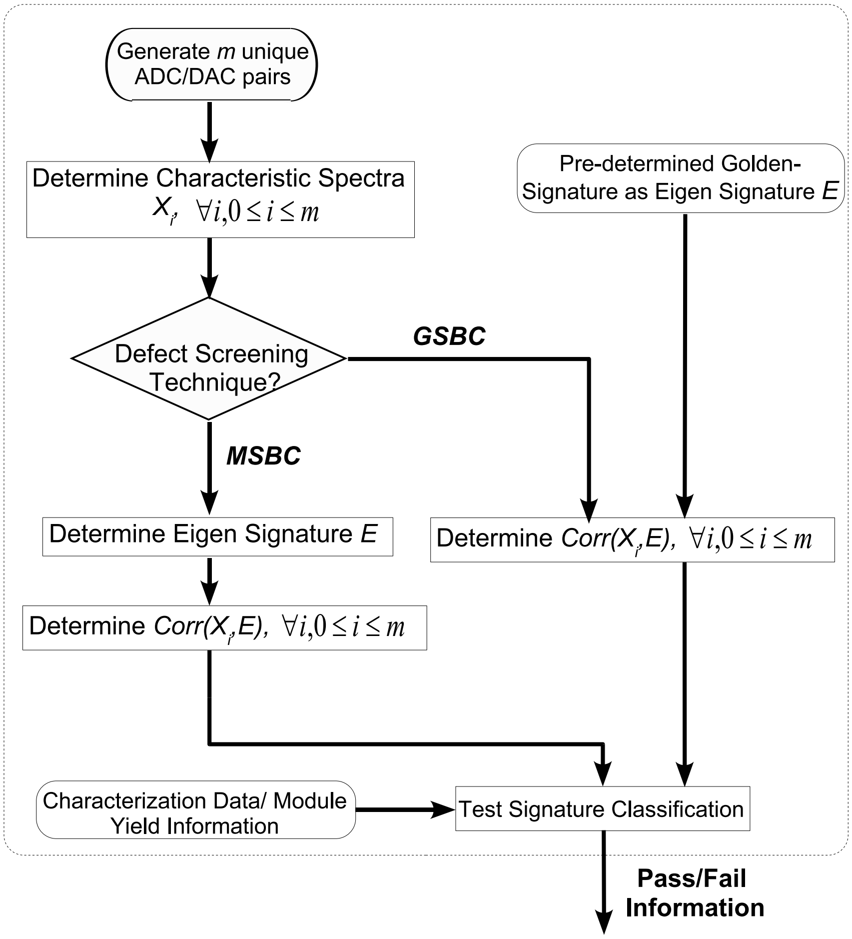


Figure 3.1: Flowchart depicting the mixed-signal test process for wafer-level fault detection.

measurements, a pre-determined threshold is of little use. However, during wafer sort, characterization data on mixed-signal components is already available. The characterization data provides information on the approximate percentage of dies that are expected to pass the final test. Modular testing of SoCs can also provide information on the approximate yield per module/core in an SoC [102]. Characterization information, in conjunction with the module yield data, can be used to estimate *a priori*, the approximate number of dies that will pass the test. The yield loss due to this indirect testing method should be minimized, since yield loss affects overall cost by increasing the effective cost of silicon per unit die. The number of passing dies can be estimated by using the expected yield ($Y_{\%}$) information from the characterization data. We set the fraction of the number of dies passing the test to be $Y_{\%} + \frac{(100-Y_{\%})}{k}$. The constant k can be chosen based on the type of signature analysis technique used.

The effectiveness of the proposed methods can be established by determining the resultant yield loss and test escapes. If G represents the number of good circuits and G_{fail} the number of good circuits failing the test, then the yield loss can be estimated to be $\frac{G_{fail}}{G}$. The number of faulty circuits that pass the test (F_{pass}) can be used to calculate the test escapes as $\frac{F_{pass}}{N-G}$.

To evaluate the above performance metrics, we develop a behavioral model of a flash-type ADC in MATLAB. We generate 1500 unique circuit instances of the ADC by inducing parametric variations in the associated components and also by injecting certain hard and soft failure types. The hard failure type corresponds to catastrophic failures and the soft failure type corresponds to parametric variations that result in undesirable circuit operation. The hard faults are generated for 100 data converters by forcing resistive opens and broken lines in the comparator network. We then vary the component parameters; the values of resistors and the offset voltages of the comparators, to generate three sets of data converters. We modify the standard deviations of resistor values and offset voltages to randomly inject the soft faults. The three sets of data converters correspond to high yield (HY-90%), moderate yield (MY-75%) and low yield (LY-60%). Correlation parameters for each unique ADC are obtained for both the proposed methods and by using a 1024-point and a 4096-point FFT. In this experiment, the specification that determines the good/faulty dies is the differential-non-linearity (DNL) parameter. The acceptable range of DNL for the ADC is set to be $0 \leq DNL \leq 0.5$. Based on the random fault injection scheme, we have a number of marginally faulty dies ($0.5 \leq DNL \leq 1$), moderately faulty dies ($1 \leq DNL \leq 2$) and grossly faulty dies ($DNL > 2$). The percentages of marginal, moderate and grossly faulty data converters in the overall population are 44%, 37% and 19% respectively.

We present experimental results for the 8-bit flash ADC model in Table 3.1. It is clear that the MSBC technique outperforms the GSBC technique in most cases, both in terms yield loss (YL) and overall test escapes (OTE). Table 3.1 lists the percentage of test escapes for marginal (TE_{MaF}), moderate (TE_{MoF}), and grossly (TE_{GF}) faulty dies. The percentages are given in terms of the number of faulty dies in each group). Columns 5-7 list the relevant data separately for each fail type. As a result, the rows of the table for these three columns do not add up to 100%. This analysis is performed in order to evaluate the effectiveness of our proposed signature analysis techniques over different failure regions. A significant percentage of marginal failures result in test escapes. This shows that the proposed signature analysis technique is not effective for screening marginal failures. On the other hand, 33%–92% and 26%–92% of the moderately faulty dies are screened in the case of the MSBC technique and GSBC technique, respectively. Thus our technique is effective for screening moderate and gross failures, which is typically the objective in wafer-level testing. Marginal failures are best detected at package test, where the chip can be tested in a more comprehensive manner.

3.2 Generic cost model

In this section, we present a cost model to evaluate wafer-level testing for a generic mixed-signal SoC. A cost model for an entire electronic assembly process is described in [103], using the concept of “yielded cost”. However, it cannot be readily adapted for wafer-level testing. In [27], a cost modeling framework for analog circuits was proposed, but it did not explicitly model the precise relationship between yield loss, test escape and the overall product cost. The effects of yield loss and test escape for

Table 3.1: Wafer-level defect screening: experimental results for an 8-bit flash ADC.

Correlation Technique	FFT: No. of Sample Points-Yield Type	YL (%)	OTE (%)	TE_{MaF} (%)	TE_{MoF} (%)	TE_{GF} (%)	k
Mean Signature	1024-LY	0.8176	46.66	89.25	33.21	3.53	5
	1024-MY	0.25	67.7	97.11	66.19	0	7
	1024-HY	0.9	49	95.23	54	7.4	7
	4096-LY	0.06	47	77.1	7.95	0	10
	4096-MY	0.08	27.43	58.65	12.67	0	10
	4096-HY	0	25	95.23	10	0	10
Golden Signature	1024-LY	1.006	75.71	98.59	73.7	42.47	5
	1024-MY	0.0375	68.75	96.15	67.6	5	7
	1024-HY	1.1	74	100	76	55.55	5
	4096-LY	0.18	29.78	88.31	7.95	0.88	8
	4096-MY	0.16	43.36	96.15	15.49	0	10
	4096-HY	0.1	2.5	100	8	0	10

YL → Yield loss; OTE → Overall test escapes; TE_{MaF}, TE_{MoF} and TE_{GF} → Test escapes for marginal, moderate and grossly faulty dies respectively.

both the digital and mixed-signal cores in an SoC is modeled in our unified analytical framework. The proposed model also considers the cost of silicon corresponding to the die area.

3.2.1 Correction factors : Test escapes and yield loss

Testing at the wafer level leads to yield loss and test escapes. Yield loss occurs when testing results in the misclassification of good dies as being defective, and the dies are not sent for packaging. We use the term Wafer-Test-Yield Loss (WYL), to refer to the yield loss resulting from wafer-level testing, and the associated non-idealities. Clearly, WYL must be minimized to reduce product cost.

The test escape component is also undesirable, due in large part to the mandated levels of shipped-product quality-level (SPQL), also known as defects per million, which is a major driver in the semiconductor industry. SPQL is defined as the fraction of faulty chips in a batch that is shipped to the customer. Test escapes at the wafer-level are undesirable because they add to packaging cost, but they do not increase SPQL if these defects are detected during module tests.

In order to make the cost model robust, we introduce correction factors to account for the test escapes and WYL. The correction factor for test escapes is obtained from the "fault coverage curve", which shows the variation of the fault coverage versus the number of test vectors. It has been shown in [2], and more recently in [104], that, the fault coverage curve can be mapped to a log function of the type $fc_n = 1 - \alpha e^{-\beta n}$, where n is the number of test patterns applied, fc_n is the fault coverage for n test patterns, α and β are constants specific to the circuit under test and the fault model used.

Typically in wafer-level testing for digital cores, only a subset of patterns are applied to the circuit, i.e., if the complete test suite contains n patterns, only $n^* \leq n$ patterns are actually applied to the core-under-test. The correction factor θ_{n^*}, defined as $\theta_{n^*} = \frac{(fc_n - fc_{n^*})}{fc_n}$, $0 \leq n^* \leq n$, is used in the model to account for test escapes during wafer-level testing.

Figure 3.2 shows how the fault coverage varies as a function of the number of applied test vectors for the digital portion of a large industrial ASIC, which we call Chip K[1]. The digital logic in this chip contains 2,821,647 blocks (including approximately 334,000 flip-flops), where a block represents a cell in the library. The figure also shows the correction factor as a function of the number of test vectors applied to the same circuit. Section 3.1 showed how we can evaluate the test escapes for analog cores. Let us assume that the test escape for analog cores is β. Assuming that test escapes for the analog cores are independent from the test escapes for digital cores (a reasonable assumption due to the different types of tests applied for the two cores), the SoC test escape can be estimated to be $1 - (1 - \theta_{n^*}) \cdot (1 - \beta)$.

Let us now consider the correction factor due to WYL. If the WYL for the digital part of the SoC is WYL_d and that for the analog part of the SoC is WYL_a, the effective WYL for the SoC is simply given by $WYL_{eff} = 1 - (1 - WYL_d) \cdot (1 - WYL_a)$.

---

[1] ASICs Test Methodology, IBM Microelectronics, Essex Jct, VT 05452.

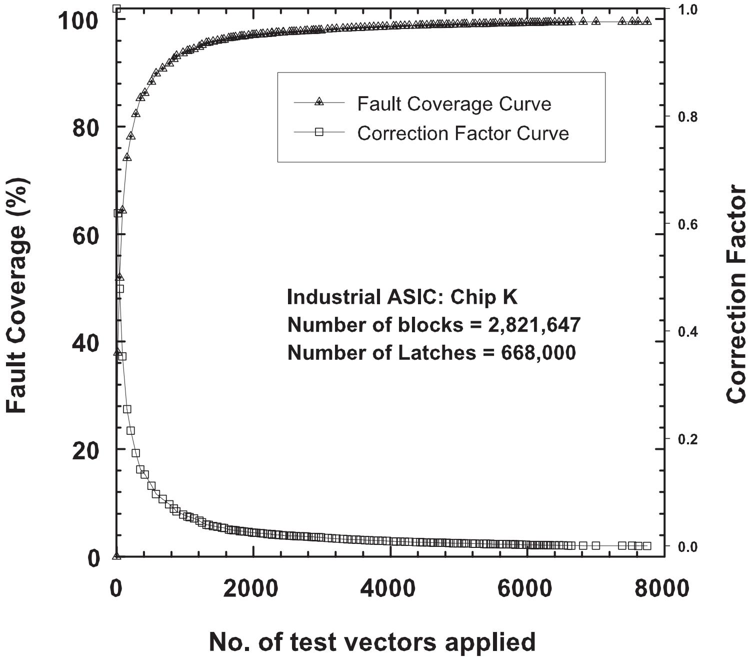


Figure 3.2: The variation of the fault coverage and correction factor versus the number of test vectors applied to the digital portion of Chip K.

The parameter WYL_d can be negligible if overtesting, which is a major concern nowadays for production testing of digital circuits [105], is not significant at the wafer-level. However, the parameter WYL_a cannot be neglected for the reasons described in Section 3.1.

3.2.2 Cost model: Generic framework

We now present our generic cost model. The cost model treats the outcomes of a test as random variables and assigns probabilities to the different possible outcomes. Appropriate conditional probabilities are used to ensure that the model takes all possible scenarios into account. Let us first define the following events: T^+: the event that the test passes, i.e., the circuit is deemed to be fault-free; T^-: the event of the test fails, i.e., the circuit is deemed to be faulty; D^+: the event that the die is fault-free; D^-: the event that the die is faulty.

Conditional probabilities associated with the above events help us to determine the various factors that influence overall test, packaging and silicon cost. The following conditional probabilities are of interest—$P(T^+ \mid D^-)$: Probability of a test pass for a faulty die (representative of test escapes); $P(T^+ \mid D^+)$: Probability of a test pass for a good die (correct classification of a good die); $P(T^- \mid D^-)$: Probability of a test fail for a bad die (correct defect screening); $P(T^- \mid D^+)$: Probability of a test fail for a good die (representative of WYL).

Using the above conditional probabilities, we can derive the following expressions for $P(T^+)$ and $P(T^-)$:

$$P(T^+) = P(T^+ \mid D^+)P(D^+) + P(T^+ \mid D^-)P(D^-) \tag{3.2}$$

$$P(T^-) = P(T^- \mid D^+)P(D^+) + P(T^- \mid D^-)P(D^-) \tag{3.3}$$

where, $P(T^+) = 1 - P(T^-)$.

$P(T^+ \mid D^-)$ denotes the test escape, while $P(T^- \mid D^+)$ indicates the yield loss. Note that $P(D^+)$ represents the yield Y of the process and $P(D^-) = 1 - P(D^+)$. Knowing these parameters, we can calculate $P(T^-)$ using Equation (3.3). Solving for $P(T^+ \mid D^+)$ from the above equations, we get:

$$P(T^+ \mid D^+) = (1 - P(T^-) - (P(T^+ \mid D^-)P(D^-))/P(D^+) \tag{3.4}$$

The probability $P(T^+)$ represents the fraction of the total number of dies that need to be packaged. The conditional probability $P(T^+ \mid D^+)$ represents the number of good dies that are packaged i.e., it represents the fraction of dies for which the test passes when the die is fault-free. This conditional probability, which can be easily calculated using Equation (3.4), is used to calculate the effective cost per unit die from the overall test and manufacturing costs.

3.2.3 Overall cost components

The overall production cost depends on whether only after-package testing is carried out, or if wafer-level testing is done in addition to production testing. We first determine the cost when only after-package testing is carried out. Let the total number of dies being produced be N, let t_{ap} represent the total test application time at the production level and c_{ap} represent the cost of test application (in \$) per unit time during after-package testing. Let C_P denote the cost of packaging per unit die, A_{die} be the area of the die under consideration, and C_{sil} be the cost of silicon (in \$) per unit area of the die. The overall production cost $C_{oc_{ap}}$ (that includes test time cost and silicon area cost, but ignores other cost components not affected by the decision to do wafer-level testing) associated with manufacturing a batch of N dies can now be determined using Equation (3.5):

$$C_{oc_{ap}} = (N \cdot t_{ap} \cdot c_{ap}) + N \cdot C_P + (N \cdot A_{die} \cdot C_{sil}) \tag{3.5}$$

Similarly the overall cost ($C_{oc_{wap}}$) associated with the manufacture of a batch of N dies for which both wafer-level and after-package testing are performed can be determined using Equation (3.6).

$$\begin{aligned} C_{oc_{wap}} &= (N \cdot t_w \cdot c_w) + P(T^+) \cdot N \cdot C_P + (P(T^+) \cdot \\ &\quad N \cdot t_{ap} \cdot c_{ap}) + (N \cdot A_{die} \cdot C_{sil}) \end{aligned} \tag{3.6}$$

In Equation (3.6), t_w and c_w represent the overall test time at the wafer-level and the tester cost per unit time, respectively. Recall that $P(T^+)$ represents the fraction of dies that pass the test at the wafer-level. This is an indicator of the number of dies to be packaged and tested at the production level. The cost per unit die by performing wafer and production level tests ($C_{die_{wap}}$) can be calculated from Equations (3.5) and (3.6) as $C_{oc_{wap}}/(N \cdot Y \cdot P(T^+ \mid D^+))$. When only production level tests are performed the cost per unit die can be estimated to be $C_{oc_{ap}}/(N\cdot Y)$. This estimate of the cost per unit die is overly optimistic because we assume that there is no yield loss or test escape associated with after-package testing. This is usually not the case in practice. We can now define the cost savings as $(\delta C = C_{oc_{ap}}/(N \cdot Y)) - (C_{oc_{wap}}/N \cdot Y \cdot P(T^{+} \mid D^{+}))$, which indicates the reduction in production cost per die due to the use of wafer-level testing.

3.3 Cost model: Quantitative analysis

In this section, we use the model to validate the importance of wafer-testing from a cost perspective. In order to use the cost model, we need realistic values of the cost components used in the model. For this purpose, we model the section of flattened digital logic (as explained in Section 3.2) as a single core, and use relevant information from a commercial mixed-signal SoC, Chip U^{2}[2]. The mixed-signal SoC includes a pair of complementary data converters of identical bit-resolution. The data converters can be configured in such a way that each DAC is routed through the ADC for purposes of test (as explained in Section 3.1). It is appropriate to assume that the ADC and the DACs are tested as pairs because a single point of failure is a sufficient criterion to reject the IC as being faulty.

In [29], the importance of packaging is highlighted with realistic numbers on the cost of silicon and cost of packaging per unit die. Furthermore, [31, 106] provides actual packaging costs for various types of packages. In this section, we choose the cost of packaging per die after carefully studying the published data. The package cost is varied from \$1 per die to \$9 per die, which is considerably lower than published data. Lower values of package costs are considered for smaller dies. Since the cost model for wafer-level testing will predict more cost savings for higher package costs, we

---

[2] ASICs Test Methodology, IBM Microelectronics, Essex Jct, VT 05452.

choose lower values for the package cost to ensure that there is no bias in the results. Packaging costs for a high-end IC can be as high as \$100 per die [29, 106, 1]. The cost of silicon from [29] is estimated to be \$0.1 per unit mm^2. We consider three typical die sizes from industry (10mm^2, 40mm^2 and 120mm^2) corresponding to small, medium and large dies, for purposes of simulation. We use a typical industry "yield curve"[1], shown in Figure 3.3, to illustrate the spread in cost savings than is achieved by testing mixed-signal SoCs at the wafer level. The points on the yield curve correspond to the probability that the yield matches the corresponding point on the x-axis. The yield curve is appropriately adjusted to reflect distributions corresponding to die sizes, because, higher yield numbers are optimistic for large dies, and vice versa [107].

3.3.1 Cost model: Results for ASIC chip K

Test costs typically range from \$0.07 per second for an analog tester to \$0.03 per second for a digital tester[1]. The cost is further reduced dramatically for an old tester, which has depreciated from long use to a fraction of a cent per second. The proposed wafer-level test method benefits from lower test time costs, hence to eliminate any favorable bias in our cost evaluation, we assume that the test time cost is an order of magnitude higher, i.e., \$0.30 per second.

We model the test escapes by assuming that the the digital portion ASIC Chip K is tested with 4046 test patterns, and for which the test escape correction factor is calculated from Figure 3.2. The analog test time is modeled by assuming that the data converter pair is tested with a 4096-point FFT. The test escape of the mixed-signal portion of the chip is assumed to be 50%.

Figures 3.3–3.5 illustrate the effect of varying packaging costs on δC for small and large dies, respectively. The cost savings per die are analyzed for each point in the discretized yield curve. This is done in order to illustrate the spread in cost saving that can be achieved in a realistic production environment. It is evident that the savings that can be achieved by performing wafer level tests is significant, and that it decreases with increase in yield.

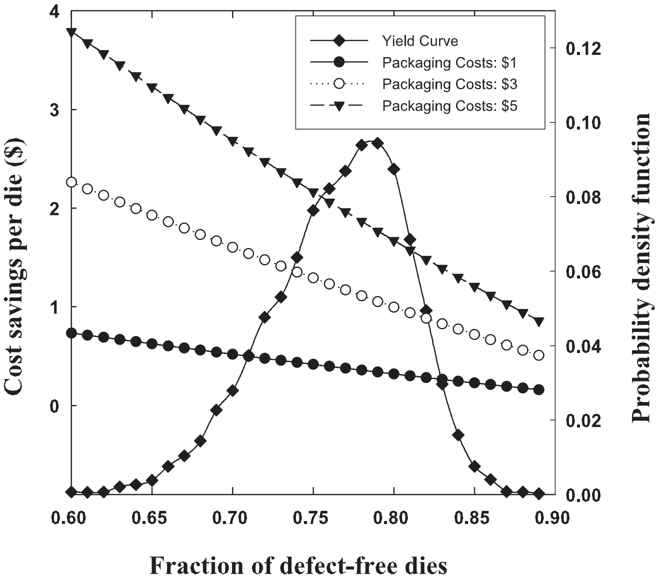


Figure 3.3: Distribution of cost savings for a small die with packaging costs of (a) \$1 (b) \$3 (c) \$5.

3.3.2 Cost model: Results considering failures due to both digital and mixed-signal cores

Until now, we have only considered chip failures that can be attributed to either the digital logic or the analog components, but not both. We next evaluate the cost savings when the digital and the analog fails are correlated. Let A denote the event of a mixed-signal test escape and B denote the event of a digital test escape. A test escape in either the mixed-signal portion of the die, or in the digital portion of the die will result in the part being packaged. The probability that the test process results in at least one test escape can be given as $P(A \cup B)$; this probability can be represented

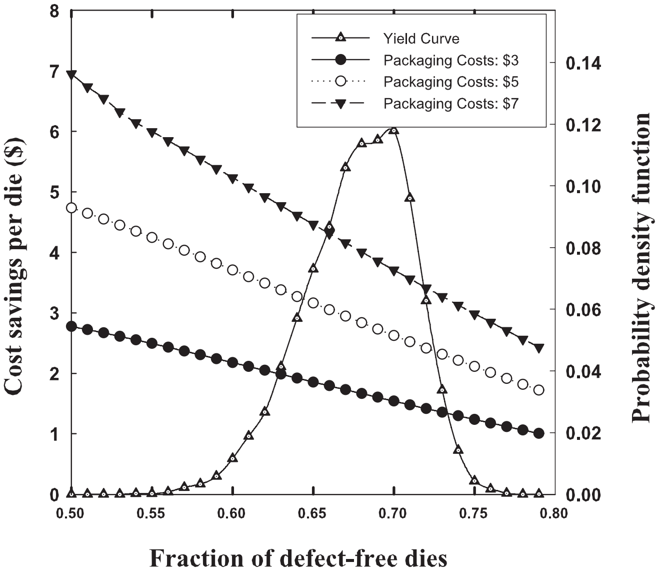


Figure 3.4: Distribution of cost savings for a medium die with packaging costs of (a) \$3 (b) \$5 (c) \$7.

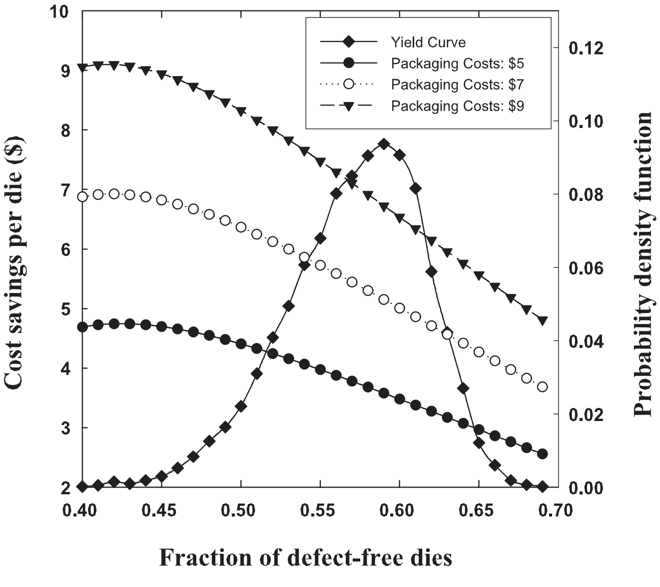


Figure 3.5: Distribution of cost savings for a large die with packaging costs of (a) \$5 (b) \$7 (c) \$9.

using the following equation:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \tag{3.7}$$

Using previously introduced notation, the above equation can be rewritten as follows:

$$P(T^{+} \mid D^{-}) = \beta + \theta_{n^{*}} - P(A \cap B) \tag{3.8}$$

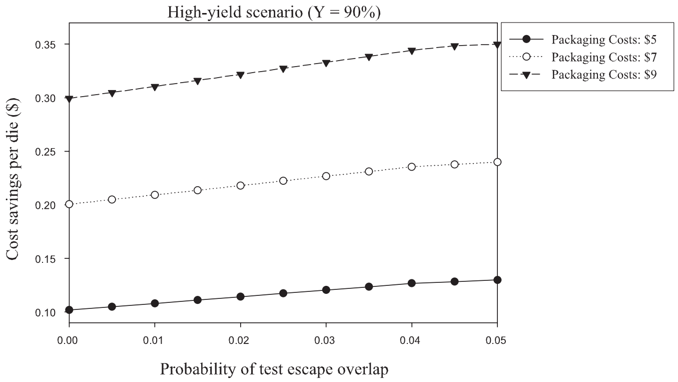
Our initial experiment considered a scenario where we assumed the test escapes occurring in the different sections of the die to be independent. We therefore took the product of the individual test escape probabilities to determines the resultant test escape. We now consider an additional scenario where test escapes occur in both parts of the die simultaneously, i.e., when a test results in a test escape in the digital portion of the die, the mixed-signal test also results in a test escape. This is given by the probability $P(A \cap B)$. In our experiments we consider test escape values by varying $P(A \cap B)$ between 0 and $\min\{P(A), P(B)\}$ to determine the test escape probability from Equation (3.8). The values of $P(A)$, $P(B)$, and the various costs associated with the test and packaging process remain the same from our experiments in Section 3.3.1. The purpose of this experiment is to determine the impact, on the overall cost savings, of test escapes that occur in both the digital and mixed-signal portions of the die.

We now present experimental results for a large die under three different yield scenarios: high yield (Figure 3.6(a)) where the yield is 90%, medium yield (Figure 3.6(b)) where the yield is 75%, and low yield (Figure 3.6(c)), where the yield is 60%. The x-axis denotes the probability of test escape overlap; the overlap in test escape is varied from 0 to the test escape probability of digital cores (0.05 for Chip U). It is observed from Figure 3.6 that our defect screening technique results in cost savings despite the overlap in test escapes in the digital and mixed-signal cores. The cost savings are minimum when there is no overlap in test escapes between the digital and mixed-signal cores, and vice-versa. Similar results are obtained for small and medium dies.

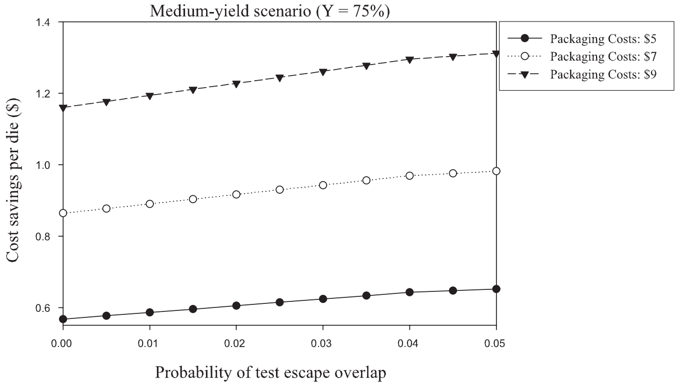
3.3.3 Cost model: Results considering failure distributions

The results in Figures 3.3–3.5 do not consider the breakdown between the various mixed-signal fail types. The percentage of marginal, moderate and gross failures can be determined via statistical binning of failure information for a given batch of dies being manufactured. Unfortunately, such failure data is not easily available in the literature; companies are reluctant to disclose this information. Therefore, we consider different scenarios and a range of values for the percentages corresponding to the different failure types. Let x_1, x_2 and x_3 represent the percentage of failures corresponding to marginal, moderate and gross fail types, and TE_1, TE_2 and TE_3 be their corresponding test escape rates. The test escape (β) for the analog cores can now be calculated as: $(TE_1 \cdot x_1 + TE_2 \cdot x_2 + TE_3 \cdot x_3)/(x_1 + x_2 + x_3)$.

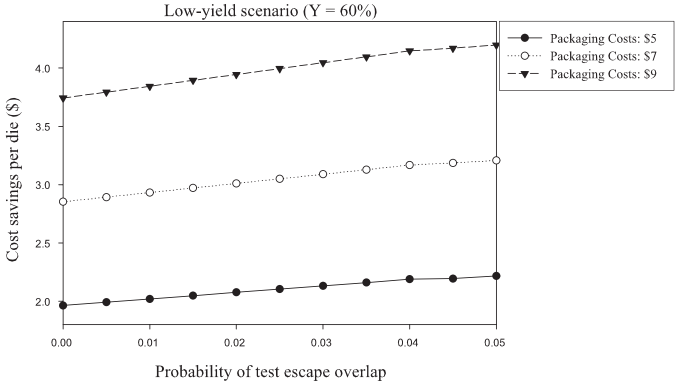
We first consider the following cases: 1) all the fail types are equally distributed; 2) the marginal fail type dominates the sample fail population; 3) the moderate fail type dominates the sample fail population; 4) the gross fail type dominates the sample fail population. In the case of a particular fail type dominating the sample fail population, we assume that the other two fail types make equal contributions to the number of failing dies. Table 3.2 illustrates the above four cases; it is assumed here that the digital core in the SoC is tested with 4046 digital test patterns. The packaging costs are chosen according to the yield type considered. We consider a packaging cost of \$5 for the low yield case, since the low yield case nominally corresponds to large dies. Similarly we consider packaging costs of \$3 and \$1 for the medium and high yield cases respectively. The die areas considered are, $10\text{mm}^2, 40\text{mm}^2$ and 120mm^2,



(a) Low-yield scenario



(b) Medium-yield scenario



(c) High-yield scenario

Figure 3.6: Distribution of cost savings for a large die with packaging costs of (a) $5, (b) $7 (c) $9, when test escapes between digital and analog parts are correlated.

corresponding to low, medium and high yield. A constant yield loss of 1% for all test cases is considered. The percentage test escapes corresponding to failure type are determined from Table 3.1 for all yield cases. The choices of packaging costs reflect the lower bounds from the values considered in Section 3.3.1. We assume here that the digital and mixed-signal fails are uncorrelated, due to the lack of representative information. In practice, as discussed in Section 3.3.2, the correlation information can be easily incorporated in the cost model if it is available for failing dies.

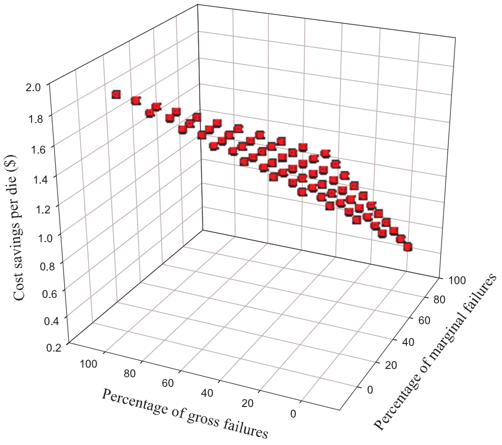
Table 3.2 presents results obtained using the cost model for the different cases described above. We present results for both the MSBC- and the GSBC-based techniques. The purpose of this experiment is to relate the importance of the proposed wafer-level defect screening techniques to the dominance of a particular fail type. It is obvious that a sample population with a high marginal fail type will result in a high overall test escape rate for the SoC (TE_{MSBC} and TE_{GSBC}). On the other hand, the test escape rate will be low for the gross fail type. Table 3.2 shows that irrespective of the distribution of fail types, wafer-level testing reduces cost in most cases. The use of the MSBC-based technique results in greater cost savings (CS_{MSBC}), compared to the GSBC technique (CS_{GSBC}). For a process known to have high yield, wafer-level testing does not always reduce test and packaging costs. The negative entries in Table 3.2 provide a reality check on the extent to which wafer-level tests should be applied. These results help us to judiciously determine the extent of wafer testing for different scenarios. The GSBC technique is inefficient for testing in a high-yield production environment, which typically corresponds to the manufacture of small dies. It is more suitable for low- and medium-yield dies.

We next vary x_1, x_2, and x_3, each between 0 and 100, under the constraint that $x_1 + x_2 + x_3 = 100$. The resulting cost savings are shown in Figure 3.7. The three axes in Figure 3.7 denote the percentage of marginal failures (x_1), the percentage

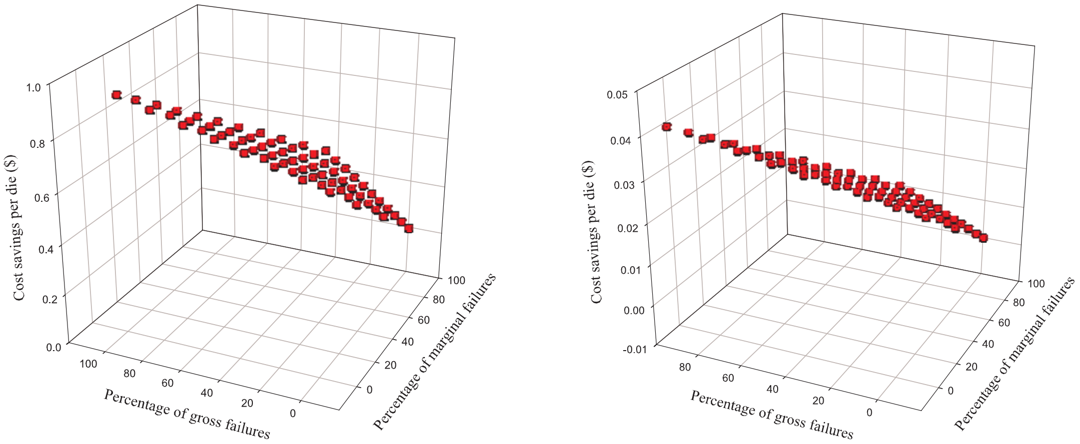
Table 3.2: Experimental Results for Cost Savings Considering Failure Type Distributions for Mixed-Signal Cores.

Yield Type	Distribution: {Marginal, Moderate, Gross} Failures (%)	FFT: No. of Sample Points	TE_{MSBC} (%)	CS_{MSBC} (in $)	TE_{GSBC} (%)	CS_{MSBC} (in $)
Low Yield (60%)	{33.33,33.33,33.33}	1024	42.79	1.6867	72.01	0.702
		4096	29.3	2.1333	32.99	2.009
	{70,15,15}	1024	68.42	0.8227	86.65	0.2084
		4096	55.78	1.24	63.65	0.9744
	{15,70,15}	1024	38.02	1.8472	73.26	0.6597
		4096	18.27	2.5057	20.45	2.4454
	{15,15,70}	1024	21.92	2.3898	56.21	1.2343
		4096	13.95	2.6513	16.26	2.5733
Medium Yield (75%)	{33.33,33.33,33.33}	1024	55	0.3867	56.79	36.86
		4096	24.82	0.685	38.04	0.5509
	{70,15,15}	1024	78.21	0.1519	78.49	0.149
		4096	43.74	0.4932	70.04	0.2265
	{15,70,15}	1024	61.44	0.3216	63.01	0.3051
		4096	18.79	0.746	26.29	0.67
	{15,15,70}	1024	25.53	0.6848	29.05	0.6492
		4096	11.92	0.8157	17.89	0.7552
High Yield (90%)	{33.33,33.33,33.33}	1024	52.81	0.0258	76.73	-0.0011
		4096	35.93	0.0381	59.96	0.018
	{70,15,15}	1024	76.2	-0.0005	89.87	-0.0159
		4096	68.6	0.001	82.25	-0.0145
	{15,70,15}	1024	53.84	0.0247	76.85	-0.0013
		4096	22.36	0.0535	71.4	-0.0021
	{15,15,70}	1024	28.56	0.0532	65.76	0.0113
		4096	16.94	0.0596	28	0.0471

TE_{MSBC} and TE_{GSBC} → Overall test escape rate for the SoC using MSBC and GSBC;
CS_{MSBC} and CS_{GSBC} → Cost savings per die using MSBC and GSBC.



(a) Low-yield scenario



(b) Medium-yield scenario

(c) High-yield scenario

Figure 3.7: Variation in cost savings considering the impact of mixed-signal fail types.

of gross failures (x_3), and the cost savings per die, respectively. (The percentage of moderate failures, x_2, is derived from x_1 and x_3.) Results are presented for three different yield scenarios: low, medium, and high yield; MSBC is used as the defect-screening technique for the results presented in Figure 3.7. It is observed that the cost savings are the least when marginal failures dominate the fail population. Similarly, a fail population with significant gross failures result in high cost savings per die. As expected, the cost savings for moderate failures lies between the cost savings for marginal and gross failures. Similar results are observed when GSBC is used instead of MSBC.

3.4 Summary

We have proposed a wafer-level defect screening technique for core-based mixed-signal SoCs. Two new correlation-based signature analysis methods have been presented for wafer-level testing of analog cores. A comprehensive cost model has been developed for a generic mixed-signal SoC; this model allows us to quantify the savings that result from wafer-level testing. Test escape, yield loss, and packaging have been incorporated in this production cost model. We have used an industrial mixed-signal SoC to evaluate the proposed wafer-level test method. The proposed method uses a low-cost digital tester for wafer-level mixed-signal test, which further reduces test cost.

The next chapter presents a test scheduling technique for WLTBI of core-based SoCs. The objective of the proposed test-scheduling technique is to minimize the variation in power consumption during WLTBI, while maintaining a reasonable test application time for the SoC.

Chapter 4

Wafer-Level Test During Burn-In (Part 1): Test Scheduling for Core-Based SOCs

Test scheduling of core-based SoCs leads to varying junction temperatures during test application. This is due to the varying power consumption of the multiple heterogeneous cores that are tested in parallel. This can result in a device being subjected to excessive or insufficient burn-in, and in certain cases may result in thermal runaway.

In this chapter, we present a power-conscious test-scheduling technique for WLTBI of core-based SoCs [108]. This technique allows us to select cores that are tested in parallel while minimizing the overall variation in power. Minimizing the overall variance in power results in less fluctuations in the junction temperature of the device. Test scheduling for WLTBI of core-based SoCs is important because of the following reasons:

1. Scheduling the cores serially during WLTBI does not satisfy the objectives of dynamic burn-in. The objective of dynamic burn-in is to have the maximum switching activity, so that all the latent defects can be screened efficiently. Testing a single core at a time does not contribute significantly towards stressing the device.
2. Even though the burn-in time is long, all the time is not allocated for test purposes. Burn-in involves temperature and voltage cycling multiple times [40]. Burn-in also involves subjecting the device to a period of static burn-in when no patterns are applied. Any minimization in test time that can be achieved

through a test scheduling technique will help minimize the overall time required for WLTBI, while at the same time satisfying the twin objectives of burn-in and test.

3. All the die in the wafer cannot be contacted during WLTBI [6]. This can be because of the lack on sufficient probe pins and/or limitations of WLTBI equipment to remove heat. When only a fraction of the die can be tested during burn-in, it is important to have low test times for the SoCs in order to test all die during WLTBI.

The main contributions of this chapter are as follows:

- We motivate the importance of handling thermal problems during WLTBI from a test-application perspective, and show how test scheduling can be used to alleviate these problems.
- We formulate a test-scheduling problem for WLTBI of core-based SoCs. Our goal is to minimize the variations in the test power of the SoC during test application
- We prove that the test-scheduling problem for WLTBI is NP-complete. We develop a heuristic technique to solve the test-scheduling problem for core-based SoCs.

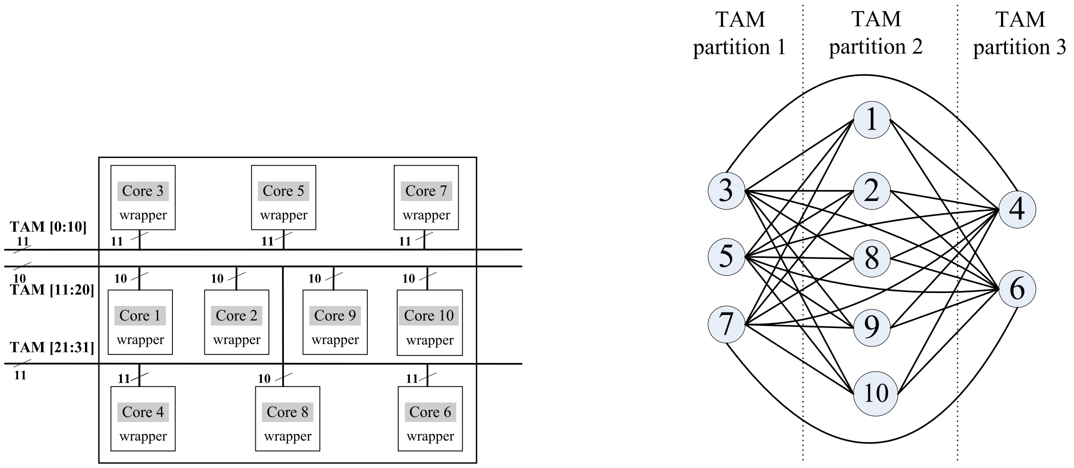
The remainder of this chapter is organized as follows. Section 4.1 formulates the test-scheduling problem for WLTBI. The heuristic method to solve the problem efficiently is presented in Section 4.2. Section 4.3 presents the baseline methods. The simulation results for three off the ITC'02 SoC benchmarks are presented in Section 4.4. Finally, we summarize the chapter in Section 4.5.

4.1 Test scheduling for WLTBI

Efficient test-scheduling methods target increased test concurrency to reduce the test application time. This leads to increased power consumption during test. Recent test-scheduling techniques for core based SoCs have included the additional dimension of test power consumption [109, 17]; this ensures that a pre-determined limit on power consumption is not exceeded during test. These techniques, however, do not address the variations in power that occur during test application. We develop a power-conscious test scheduling approach in this chapter, tailored for WLTBI of core-based SoCs. The primary objective of our work is to minimize variations in power consumption such that predictions on burn-in time are accurate. A secondary objective is to minimize the test application time.

4.1.1 Graph-matching-based approach for test scheduling

A graph $G(V_1, \cdots, V_N; E)$, where $V_1, V_2, \cdots, V_N$ are subsets of vertices and E is the set of edges, is B-partite if there is no edge between any two vertices in the vertex subset V_i, $1 \leq i \leq N$ [110]. In other words, the vertices are partitioned into B sets (partitions), such that no two vertices contained in any one partition are adjacent to one another. The assignment of cores to TAMs in an SoC can be represented by a complete B-partite graph, where B is the number of TAM partitions; the set of vertices, V_i denotes the cores assigned to TAM partition i. The edges between the nodes in the different partitions model the fact that at any clock cycle, any group of cores on different TAM partitions are candidates for concurrent testing. An example of a TAM architecture for the d695 SoC with a TAM width $W = 32$ is shown in Figure 4.1(a). This can be represented by a complete B-partite graph ($B = 3$) as shown in Figure 4.1(b); this is also known as a complete tripartite graph. Edges exist between all pairs of nodes (cores) in different partitions; this implies that at



(a) (b)

Figure 4.1: (a) TAM architecture for the d695 SoC with $W = 32$ (b) Corresponding B-partite ($B = 3$) graph, also referred to as a tripartite graph for the d695 SoC with $W = 32$. The nodes correspond to cores.

any given clock cycle, any set of cores on the three different TAM partitions can be tested concurrently.

We assume a fixed-width TAM architecture and test buses [111], where the division of W wires into B TAM partitions has been determined *a priori* using methods described in [111, 84]. We now have to determine an optimal ordering of cores such that the overall variation in power consumption for the SoC is minimized while satisfying the constraint on peak power consumption P_{max}. We refer to this problem as $\mathcal{P}_{Core\_Order}$. We use the following two measures as metrics to analyze the variation in power consumption.

1. The first measure is the statistical variance in test power consumption. Let T_{SoC} represent the test time for the SoC in clock cycles, and P_{mean} the mean value of power consumption per clock cycle during test. The variance in test power consumption for the SoC is defined as $\frac{1}{T_{SoC}} \sum_{i=1}^{T_{SoC}} (P_i - P_{mean})^2$. Low variance indicates low (aggregated) deviation in test power from the mean value of power consumption during test. Successful WLTBI requires the minimization of this

metric.

2. The cycle-to-cycle variation in test power consumption is an indicator of the "flatness" of the power profile during test. Large cycle-to-cycle power variations are undesirable. We therefore quantify the "flatness" in the power profile using the metric $\gamma = \frac{\sum_{i=1}^{T_{SoC}-1} |P_{i+1} - P_i|}{T_{SoC}-1}$; P_i and P_{i+1}, denote the power consumption during the i^{th} and $(i+1)^{th}$ clock cycles. Low values of γ are desirable for WLTBI.

Without loss of generality and to simplify the presentation, we henceforth consider an SoC with three TAM partitions ($B = 3$). (The extension to more than three TAM partitions is straightforward.) The problem $\mathcal{P}_{Core\_Order}$ for an SoC with three TAM partitions can now be formally stated as follows:

Problem $\mathcal{P}_{Core\_Order}$: Let T_1, T_2 and T_3 be the sets of cores on TAM partitions 1, 2 and 3 respectively. Determine the sets of cores that can be tested simultaneously, and the ordering of the cores on the TAM partitions, such that the overall variation in power consumption for the SoC is minimized and the peak power constraint P_{max} is satisfied.

We relate the core-ordering problem to the maximum tripartite graph-matching problem [112]. For any three sets X, Y, Z, and a corresponding set S of triples $X \times Y \times Z$, the maximum tripartite matching problem determines a maximum matching set of triples M, $M \in S$. The elements of X, Y and Z that are matched occur exactly in one triple $\in M$, and for any other matching M', $|M| \geq |M'|$.

To solve $\mathcal{P}_{Core\_Order}$, we need to determine sets of cores that are tested (concurrently) during a test-session, and the ordering of these sets of cores in the test schedule such that the variation in power consumption during test is minimized. The tripartite graph, such as the the one shown in Figure 4.1(b), is used to represent the assignment of cores to TAMs in an SoC. A matched triple in the tripartite graph represents a set of three cores that are concurrently tested during a test-session without violating the peak power constraint P_{max}. The numbers of cores on each TAM partition in an SoC are not necessarily equal. It is therefore necessary to determine matched sets of triples, matched edges, and unmatched vertices (in the same order) iteratively for the tripartite graph, to ensure that all the cores are assigned to the test schedule. A graph-theoretic matching procedure can be used to determine and order the matched triples, the resulting matched edges, and unmatched vertices. We next describe how edge weights are added to the tripartite graph. These weights indicate the power variation during test.

In a weighted tripartite graph $\mathcal{G}$, a weight $w(e)$, is associated with each edge e. The edge weight in the context of $\mathcal{P}_{Core\_Order}$ can be used to numerically represent the variation in power consumption when the two cores corresponding to the vertices at the end-points of the edge are tested concurrently. The tripartite graph can now be augmented to include weights for groups of three vertices, one from each partition in the tripartite graph. This weight $\rho(i, j, k)$ is used to represent the variation in power consumption when the three cores i, j, and k are tested in parallel. It is given by $\rho(i, j, k) = \mu(i, j, k) + \sigma(i, j, k)$; the parameter $\mu(i, j, k)$ is the statistical mean and $\sigma(i, j, k)$ is the standard deviation in power consumption, when cores i, j, and k are tested concurrently. Note that $1 \leq i \leq |T_1|$, $1 \leq j \leq |T_2|$ and $1 \leq k \leq |T_3|$.

We next determine a matching in the tripartite graph that results in the least "cost". The cost of a matching here corresponds to the aggregate variation in power consumption when the cores corresponding to the matched groups of vertices and matched edges, are assigned to test sessions in the test schedule for WLTBI. The matching problem uses the weighted tripartite graph to determine matched sets of three vertices and matched edges in the order of increasing weight to obtain a match with the lowest cost. Matched sets of three vertices and edges in the order of increasing weight leads to a reduction in the variance in test power, and the mean cycle-to-cycle variation in test power. The least-cost weighted tripartite graph-matching problem is

Problem $\mathcal{P}_{GMP}$: Given a weighted tripartite graph $\mathcal{G}$, determine a lowest-cost matching for $\mathcal{G}$.

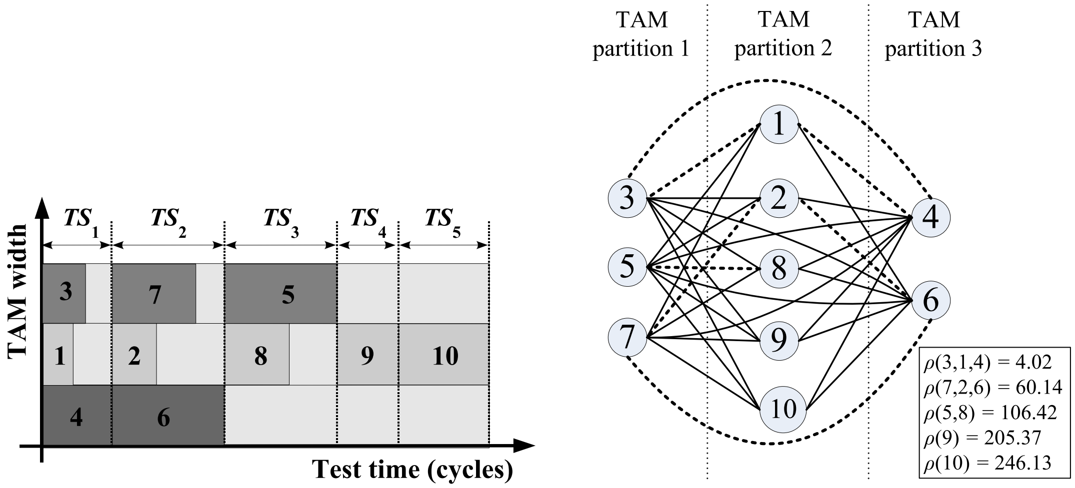
Figure 4.2(a) illustrates an example test schedule optimized for WLTBI. The first two test sessions TS_1 and TS_2 in the test schedule correspond to matched set of triples $\{3, 1, 4\}$, and $\{7, 2, 6\}$ in the weighted tripartite graph shown in Figure 4.2(b). The dotted lines in Figure 4.2(b) represent matching in the tripartite graph. Cores 3, 1, and 4 when tested concurrently result in the least power variation among all valid core combinations. The power data for this example is taken from the cycle-accurate test modeling approach presented in [109]. The test session TS_3 is represented by a matched edge $\{5, 8\}$ in Figure 4.2(b). Cores 9 and 10 represent unmatched vertices in the tripartite graph, and they are tested individually in the test schedule. The solution to $\mathcal{P}_{GMP}$ therefore corresponds to a solution for $\mathcal{P}_{Core\_Order}$.

We next use the method of restriction to prove that $\mathcal{P}_{Core\_Order}$ is NP-hard. A special case of $\mathcal{P}_{Core\_Order}$, where $|T_1| = |T_2| = |T_3| = n$ and all the edge weights are equal, is equivalent to the well-known perfect tripartite matching problem [112]. The perfect tripartite matching problem is stated as follows:

Instance: Three disjoint subsets X, Y and Z, where $|X| = |Y| = |Z| = n$, and a set of triples $S \subseteq X \times Y \times Z$.

Question: Is there a matched set of triples M, where $M \subseteq S$ such that $|M| = n$, and every element of M occurs exactly in one triple of S?

The perfect tripartite matching problem is known to be NP-Complete [112]. Since a special case of $\mathcal{P}_{Core\_Order}$ is equivalent to a general instance of the perfect tripartite



(a) (b)

Figure 4.2: (a) Test schedule for the d695 SoC with $W = 32$ and $P_{max} = 1800$. (b) Matched tripartite graph for the d695 SoC with $W = 32$. Dotted lines represent matching.

matching problem, it follows from the method of restriction that $\mathcal{P}_{Core\_Order}$ is NP-hard.

4.2 Heuristic procedure to solve $\mathcal{P}_{Core\_Order}$

We next describe the heuristic algorithm that we use to solve $\mathcal{P}_{Core\_Order}$. The algorithm starts with an initial assignment of cores to TAM partitions, and then iteratively (re)assigns cores to the three TAM partitions such that the variation in test power is minimized. The main steps, as shown in Figure 4.3, are outlined below:

1. In procedure $Initial\_Assign$, we schedule cores that are tested first on each TAM partition, i.e., their test start-times are zero. The assignment of cores is obtained by determining the set of cores that yield the least variation in power consumption when tested simultaneously.

2. In procedure $Assign\_Cores$, we determine the next sets of cores that are assigned to the test schedule. Cores are iteratively scheduled in sets of three, until all cores

```
Algorithm Core_Order
Initial_Assign();
Determine $\rho(i,j,k)_{init} = \mu(i,j,k)_{init} + \sigma(i,j,k)_{init}$;
while there is a matched triple, do
    Assign_Cores();
    delete vertices corresponding to the triple chosen
    by Assign_Cores();
end while
while $|T_1|, |T_2|$, and $|T_3| \neq \emptyset$; do
    Unmatched_Assign();
end while
return test schedule for the cores in the SoC;
```

Figure 4.3: Pseudocode for the *Core_Order* heuristic procedure.

corresponding to the matched connections in the tripartite graph are scheduled.

3. In procedure *Unmatched_Assign*, we determine the assignment of cores (vertices) that have not been matched. If all the cores in a particular TAM partition have already been scheduled, *Unmatched_Assign* selects cores from the remaining TAM partitions to reduce the overall variation in test power.

The proposed solution can be easily extended for SoCs with more than three TAM partitions. Instead of a tripartite graph, we will need a B-partite graph for B TAM partitions. The *Initial_Assign* procedure and the *Assign_Cores* procedure both require searching through N^3 candidate solutions in the worst case; hence the time complexity is $O(N^3)$, where N is the number of cores in the SoC. The worst-case time complexity of the heuristic procedure in terms of the number of TAM partitions B is $O(N^B)$. The heuristic procedure is exponential in the number of TAM partitions B, but B is a constant at wafer-level since the TAM architecture is optimized during design time for package test.

4.3 Baseline methods

We next describe two baseline methods. The first baseline method solves a power-constrained test-scheduling problem for core-based SoCs. This approach considers a single power-limit value for the entire SoC [109]. We determine the variation in power consumption over time, when only a peak power limit is considered for test scheduling. We use the same TAM architecture used by the $Core\_Order$ heuristic.

The baseline scheduling algorithm keeps a record of the per-cycle values of power consumption and ensures that it is less than P_{max} at every cycle. When a new core is added to the test schedule, the test power for the core is accumulated to reflect the overall power consumption profile of the SoC. The algorithm iteratively schedules the cores in the SoC to minimize the SOC test time, while satisfying the power limit P_{max}.

In the second baseline method, we consider a pre-designed TAM architecture, where the division of W top-level TAM wires into B TAM partitions, and the assignment of cores to these TAM partitions are determined *a priori* using methods described in [111] for package test. We then test these cores serially with their pre-allocated TAM width, such that the power consumption and the variance in power consumption are kept to a minimum. No two cores are tested concurrently.

4.4 Experimental results

In this section, we present experimental results for three SoCs from the ITC'02 SoC test benchmarks. We use cycle-accurate power data from [109]. Since the objective of $\mathcal{P}_{Core\_Order}$ is to minimize the variation in test power consumption (represented by the two metrics presented in Section 4.1) during WLTBI, we present the following results:

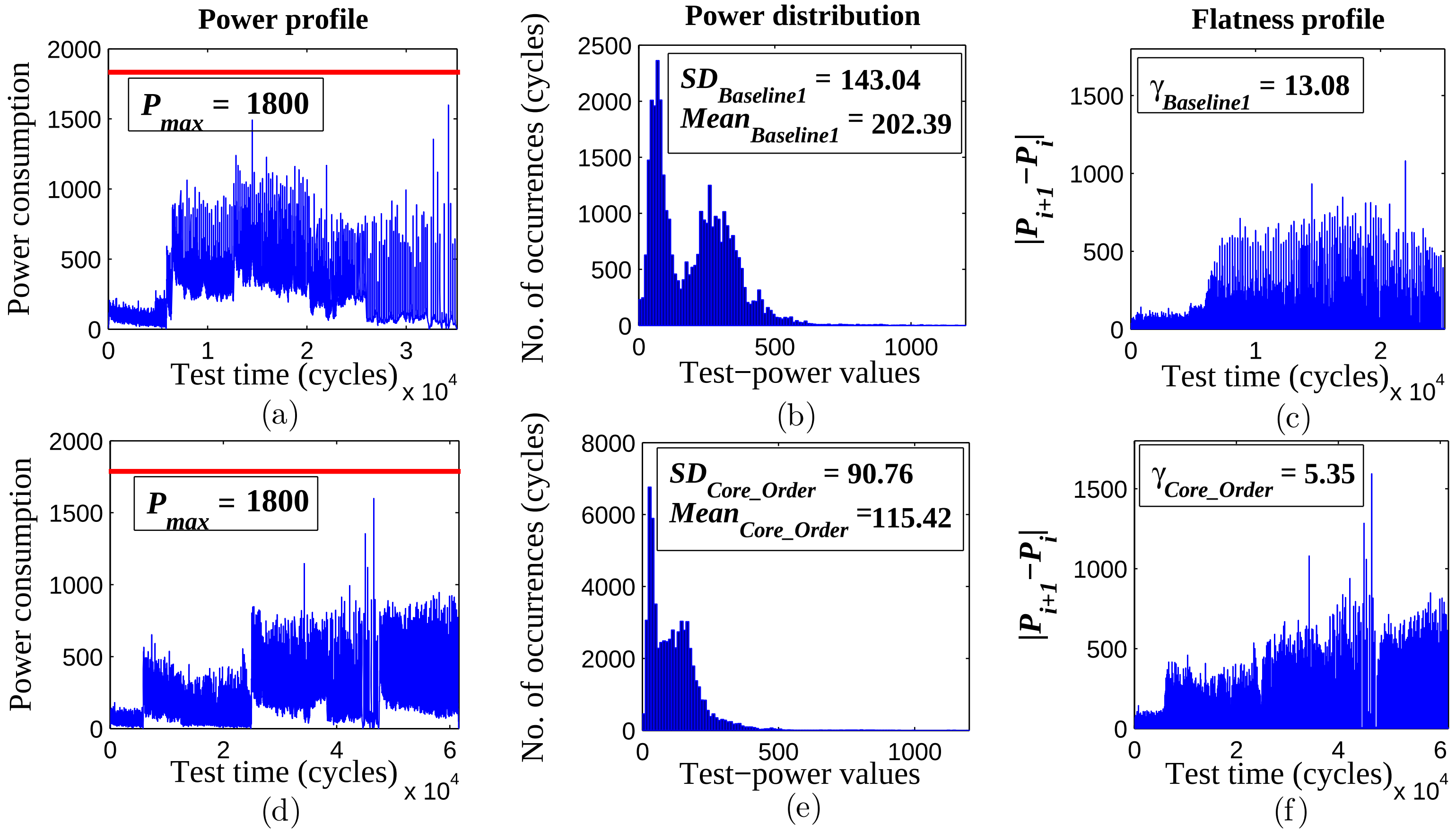


Figure 4.4: Power profile for d695 obtained using baseline approach 1 and *Core_Order* ($W = 32$ and $P_{max} = 1800$).

- The percentage difference in variance between baseline method 1 and $Core\_Order$. This difference is denoted by $\delta V_{Baseline1}$, and it is computed as $\frac{V_{Baseline1} - V_{Core\_Order}}{V_{Baseline1}} \times 100\%$; V_{Core_Order} represents the variance in test power consumption obtained using the $Core\_Order$ heuristic, and $V_{Baseline1}$ represents the variance in power consumption obtained using the first baseline method.

- The percentage difference in variance between baseline method 2 and $Core\_Order$. This is calculated in a similar fashion as $\delta V_{Baseline1}$, and is denoted as $\delta V_{Baseline2}$.

- We highlight the difference in the mean cycle-to-cycle power variation obtained using baseline method 1, and $Core\_Order$. We characterize this difference as $\delta\gamma = \frac{\gamma_{Baseline1} - \gamma_{Core\_Order}}{\gamma_{Baseline1}} \times 100\%$; $\gamma_{Baseline1}$ and γ_{Core_Order} are the "flatness" indicators obtained using the first baseline method and the $Core\_Order$ heuristic respectively.

- We also present the WLTBI test time for the SoC obtained using $Core\_Order$ and the baseline test methods.

We first present power profiles and the corresponding distribution in power consumption values during test. Figure 4.4 illustrates the power profile for the d695 SoC when tested with a TAM width of 32; the maximum value of power consumption, P_{max}, is set to 1800 units in this case. (The units are derived from [109].) Figures 4.4(a) and 4.4(b) represent the power profile during test for the baseline approach and the distribution in power consumption values corresponding to the power profile, respectively; Figures 4.4(d) and 4.4(e) represent the same information obtained using the $Core\_Order$ heuristic. Figures 4.4(c) and 4.4(f) illustrate the flatness profiles obtained for the baseline scenario and using $Core\_Order$ respectively. We can make the following observations from Figure 4.4:

- The standard deviation SD, and hence the variance in power during test, is significantly lower when $Core\_Order$ is used to determine the ordering of cores.

- The mean value of power consumption (Mean) during test is also significantly lower when the cores are ordered using $Core\_Order$. This is because $Core\_Order$ reduces the variation in power consumption at the cost of increased test time.

- The lower values of variance in power consumption obtained using the $Core\_Order$ heuristic results in a distribution where the power consumption values are packed into fewer bins in the power distribution profile as compared to the baseline approach.

- The power profile obtained using $Core\_Order$, for the case illustrated in Figure 4.4, is 59% flatter than the baseline scenario. This is an indicator of the low cycle-to-cycle power variation during test.

The results for the three benchmark SoCs, d695, p22810 and p93791 are summarized in Tables 4.1-4.3 respectively; eight different values of W are considered in each case. The values of P_{max} for each circuit are chosen carefully after analyzing the per-cycle test-power data provided in [109]. The minimum value of P_{max} is chosen such that a feasible schedule can be formulated using the given value of P_{max}. The SoC test time, TT_{Core_Order}, obtained using $Core\_Order$, and the SoC test time using the baseline cases, $TT_{Baseline1}$ and $TT_{Baseline2}$ are reported in addition to $\delta V_{Baseline1}$, $\delta V_{Baseline2}$, and $\delta\gamma$. The results show that significant reduction in test power variation can be obtained using our heuristic procedure, which ideally is the goal for WLTBI. Significant reduction in cycle-to-cycle power variation is observed for all scenarios when $Core\_Order$ is used to order the cores.

The test times for the proposed approach are higher than that for baseline method 1. Recall that test-time minimization is a secondary objective for WLTBI. The primary objective here is to minimize the test-power variance. Note that a limited increase in the test time is not a serious drawback because the wafer is subjected to relatively long intervals of burn-in.

The second baseline approach results in low values of variance for power consumption. This because the cores are tested sequentially in this case, thereby resulting in much higher test times as compared to the first baseline approach and *Core_Order*. Higher test times result in higher memory requirements; this limits the number of die that can tested in parallel during WLTBI. Temperature and voltage cycling during burn-in result in the die being tested at different operating temperatures and voltages [37]. A reasonable test time is therefore necessary to support test repetitions under such a scenario. The tester scan clock frequency for the burn-in ATE is lower than that for a conventional ATE [37]. The significantly higher test time for the second baseline method renders the method unsuitable for WLTBI.

The CPU time for test scheduling for d695 is less than a minute for all cases. The *Core_Order* procedure takes up to 2 hours for the p22810 SoC and up to 4 hours of CPU time for the p93791 SoC on a 2.4 GHz AMD Opteron processor, with 4 GB of memory.

4.5 Summary

We have formulated a test-scheduling problem for WLTBI of core-based SoCs, which minimizes the variation in test power during test application. This is the first attempt to develop a test-scheduling solution to address thermal issues that arise during WLTBI. We have used cycle-accurate test-power data for the cores to solve the test-scheduling problem. We have shown that test-scheduling under power-variation constraints can be modeled using the graph-matching problem on multi-partite graphs. We have proven that the test-scheduling problem $\mathcal{P}_{Core\_Order}$ is NP-Complete; there-

Table 4.1: Reduction in test-power variance for d695.

P_{max}	W	$\delta V_{Baseline1}$	$\delta V_{Baseline2}$	$\delta\gamma$	TT_{Core_Order} (cycles)	$TT_{Baseline1}$ (cycles)	$TT_{Baseline2}$ (cycles)
1600	8	27.77	0.16	35.19	247730	180799	290754
	16	65.49	−13.37	49.61	124402	60482	147568
	24	31.36	−24.61	13.60	71517	59329	96472
	32	59.74	−7.71	40.95	65870	53833	77113
	40	26.55	−23.29	20.37	61589	47442	75283
	48	6.74	−1.89	8.58	51274	35940	61868
	56	20.81	−25.49	25.02	42350	22569	49620
	64	12.57	−25.34	26.66	41882	21595	48740
1800	8	27.77	−2.56	35.19	239727	180799	290754
	16	56.52	−21.38	48.12	120468	60481	147568
	24	45.65	−24.61	51.47	71517	40383	96472
	32	59.74	−7.71	59.09	65870	53833	77113
	40	26.55	−23.29	5.76	61589	47442	75283
	48	6.74	−1.89	8.58	51274	35940	61868
	56	8.39	−21.36	40.85	40690	22569	49620
	64	11.71	−24.14	26.66	32499	21595	48740
2000	8	15.60	−10.86	31.10	191668	180798	290754
	16	56.52	−21.38	48.12	120468	60481	147568
	24	40.49	−24.61	55.03	71517	37370	96472
	32	59.74	−7.71	40.95	65870	53833	77113
	40	14.11	−21.01	5.76	61589	35124	75283
	48	5.46	−21.05	8.13	44167	30830	61868
	56	7.46	−16.38	43.10	34860	22423	49620
	64	4.17	−15.74	43.32	32499	18726	48740

Table 4.2: Reduction in test-power variance for $p22810$.

P_{max}	W	$\delta V_{Baseline1}$	$\delta V_{Baseline2}$	$\delta\gamma$	TT_{Core_Order} (cycles)	$TT_{Baseline1}$ (cycles)	$TT_{Baseline2}$ (cycles)
6000	8	45.73	−7.52	47.21	1974010	879724	2600613
	16	2.00	−10.52	43.13	1122870	550139	1375168
	24	18.41	−15.33	40.70	808702	420351	995305
	32	6.66	−12.09	24.15	675010	343927	834102
	40	28.81	−6.64	47.40	560079	318426	688086
	48	13.15	−4.65	41.68	547496	230457	661847
	56	38.92	−2.45	45.00	514440	210138	629568
	64	38.61	−4.99	47.51	483679	202185	605656
8000	8	44.86	−7.52	47.83	1974010	869465	2600613
	16	1.02	−11.07	53.52	1206435	463658	1375168
	24	4.54	−17.31	50.42	798816	338348	995305
	32	6.85	−4.89	39.59	681721	263638	834102
	40	26.16	−8.91	47.40	560079	250457	688086
	48	13.15	−4.65	49.74	547496	230457	661847
	56	38.92	−2.45	45.00	514440	210138	629568
	64	12.31	−13.02	45.08	475127	202185	605656
10000	8	44.86	−7.52	47.83	1974010	869465	2600613
	16	1.02	−11.07	53.52	1206435	463658	1375168
	24	15.59	−17.31	50.42	798816	338348	995305
	32	6.85	−4.89	35.85	681721	263638	834102
	40	6.85	−4.89	47.40	560079	250457	688086
	48	12.08	−3.13	55.36	547496	221241	661847
	56	31.74	−2.45	45.36	514440	208476	629568
	64	12.31	−13.02	45.08	475127	202185	605656

Table 4.3: Reduction in test-power variance for $p93791$.

P_{max}	W	$\delta V_{Baseline1}$	$\delta V_{Baseline2}$	$\delta\gamma$	TT_{Core_Order} (cycles)	$TT_{Baseline1}$ (cycles)	$TT_{Baseline2}$ (cycles)
15000	8	55.26	−0.02	29.68	8777550	4303557	10828293
	16	43.38	−0.05	6.27	4937767	1890881	5851966
	24	71.05	−0.19	32.83	2849621	1727943	3621107
	32	31.28	−0.02	27.99	2272156	1427138	2860859
	40	25.64	11.24	28.55	1600355	1152953	2017488
	48	18.27	−0.28	26.28	1494115	1028976	1720545
	56	35.10	−0.84	39.95	1185860	736604	1613826
	64	17.49	−2.49	40.01	1045983	694142	1478334
20000	8	64.82	−0.02	32.45	8777550	4103621	10828293
	16	43.38	−0.05	10.02	4937767	1890881	5851966
	24	79.58	−0.19	31.94	2849621	1727943	3621107
	32	31.28	−0.02	27.99	2272156	1427138	2860859
	40	25.64	11.24	28.55	1600355	1152953	2017488
	48	21.88	−1.21	29.43	1342911	991466	1720545
	56	35.10	−0.84	39.95	1185860	736604	1613826
	64	17.49	−2.49	40.01	1045983	694142	1478334
25000	8	64.82	−0.02	31.53	8777550	4103621	10828293
	16	43.38	−0.05	8.79	4937767	1890881	5851966
	24	79.58	−0.19	31.94	2849621	1727943	3621107
	32	31.28	−0.02	27.99	2272156	1427138	2860859
	40	31.25	6.19	28.55	1400129	1061723	2017488
	48	21.88	−1.21	29.43	1342911	991466	1720545
	56	33.39	−0.46	39.23	1124549	713561	3621107
	64	8.56	−4.59	41.62	1012164	662198	1478334

fore, we have presented a heuristic technique to solve $\mathcal{P}_{Core\_Order}$. Results for the ITC'02 SoC test benchmarks show that a significant reduction in power variation is obtained using the proposed method.

In Chapter 5, we present test-pattern ordering technique for WLTBI of full scan circuits. The objective of the test-pattern ordering problem is to minimize the variation in power consumption during test application.

Chapter 5

Wafer-Level Test During Burn-In (Part 2): Test-Pattern Ordering

Test application during burn-in at the wafer level requires low variation in power consumption during test pattern application. The issue of controlling the variation in power consumption during test is addressed in this chapter. We present two solution methods, that allow us to determine an ordering of test patterns for WLTBI. Reduced variance in test power results in less fluctuations in the junction temperatures of the device. The ordering methods presented help control the variation in power consumption during test; this will significantly lower the fluctuations in junction temperature.

The key contributions of this chapter are as follows:

- We motivate the importance of handling thermal problems during WLTBI, and show how test pattern ordering can be used to alleviate these problems.
- We present a test-pattern-ordering technique based on ILP for scan-based WLTBI. Our goal is to minimize the variations in the test power of the device during test application.
- We also develop heuristic techniques to solve the test pattern ordering problem for large circuits.

The remainder of the chapter is organized as follows. A brief overview of cycle-accurate power modeling technique for scan based circuits is presented in Section 5.1. Section 5.2 presents the ILP-based test pattern ordering technique for WLTBI.

In Section 5.3, a heuristic method to solve the problem efficiently is presented. The baseline methods used to evaluate the test pattern ordering techniques are presented in Section 5.4. Section 5.5 presents simulation results for several ISCAS'89 and IWLS'05 benchmark circuits [113]. Finally, Section 5.6 summarizes the chapter.

5.1 Background: Cycle-accurate power modeling

A significant percentage of scan cells change values in every scan-shift and scan-capture cycle. The toggling of scan flip-flops can result in excessive switching activity during test, resulting in high power consumption. It has been shown in [63] that the number of transitions of the DUT is proportional to the number of transitions in the device scan-chains. Therefore, a reduction in the number of transitions in the scan cells during test application leads to lower test power. A number of techniques have been developed to reduce the peak power and average power consumption during test by reducing the number of transitions in the scan chain [62, 114]. These techniques rely on test-pattern ordering [115, 116], scan-chain ordering [67, 117], and the use of multiple capture cycles during test application [115] to reduce the toggling of scan cells during shift/capture cycles. Segmented scan approaches [118, 65, 64] have also been used to address test power issues for industrial designs.

Scan-chain transition-count calculation

In [63], a metric known as the weighted transition count (WTC) was presented to calculate the number of transitions in the scan chain during scan shifting. It was also shown in [63] that the WTC has a strong correlation with the total device power consumption. The WTC metric can be extended easily to determine the cycle-by-cycle transition counts while applying test patterns. The knowledge of the length of the scan chains, the test pattern to be scanned in, and the initial state of the scan cells (response from previously applied test stimulus), can be used to generate cycle-accurate test power data.

We next illustrate the procedure to determine cycle-accurate power consumption. Let us consider the case of a circuit under test (CUT) with six scan cells $FF_1, FF_2, FF_3, FF_4, FF_5, FF_6$, and a test pattern $tp = (110110)$ being scanned in. Let the initial state of scan cells be $tr = (101110)$. Figure 5.1 represents the cycle-by-cycle change in the values of the scan cells when the test pattern is scanned in, and the test response is scanned out. The scan-in and scan-out of the test pattern and responses are not the only contributors to the change in values of the scan cells. It is also important to consider the transitions that occur during the capture cycle. The number of transitions that occur during the capture cycle can be calculated by determining the Hamming distance between the test stimuli and its expected test response.

Let us consider a scan chain of length n that has an initial value $ti = (ti_1, ..., ti_n)$, and a test pattern $tp = (tp_1, ..., tp_n)$ that is shifted into the scan chain. The transitions that occur during the shifting of the test pattern (and shifting out the previous state test response) can be represented as an $n \times n$ matrix T [109]. An element t_{ij} of T is 1 if there is a transition in scan cell j during clock cycle i; otherwise $t_{ij} = 0$. T can be used to calculate the total number of scan-cell transitions (a measure of the power consumption during test) during every clock cycle. During any given clock cycle i, the total number of transitions $tr(i)$ can be calculated by summing the values of all elements in row i of T; this can be expressed using the equation $tr(i) = \sum_{j=1}^{n} t_{ij}$.

For the example shown in Figure 5.1, the cycle-by-cycle number of scan-cell transitions is given by the set $\{4, 4, 4, 4, 5, 4\}$. For the test response (111100), the number of transitions that occur during the capture cycle for this example is 2. For multiple scan chains, the above calculation can simply be carried out independently for each

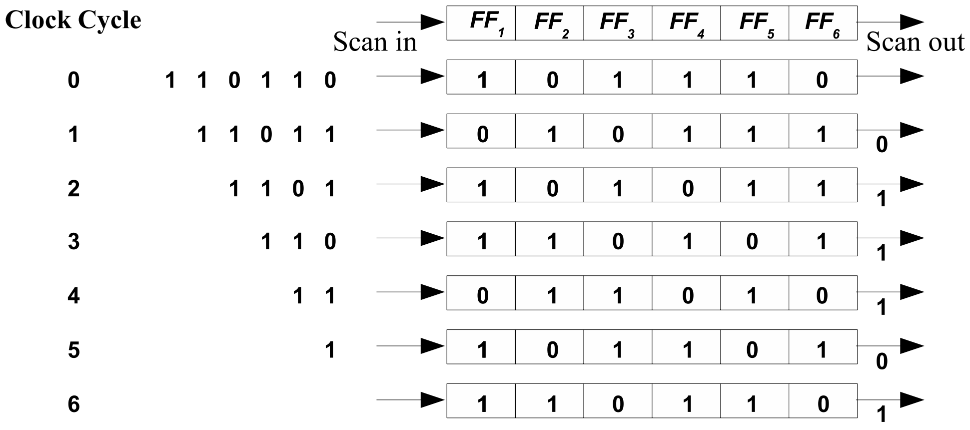


Figure 5.1: Example to illustrate scan shift operation.

scan chain.

5.2 Test-pattern ordering problem: $\mathcal{P}_{TPO}$

In this section we present the test pattern ordering problem $\mathcal{P}_{TPO}$. The goal is to determine an optimal ordering of test patterns for scan-based testing, such that the overall variation in power consumption during test is minimized. For simplicity of discussion, we assume a single scan chain for test application and N patterns $T_1, T_2, \cdots, T_N$. The extension of $\mathcal{P}_{TPO}$ to a circuit with multiple scan chains is trivial. The test application for the CUT is carried out as follows:

1. The scan flip-flops in the circuit are all assumed to be initialized to 0.

2. The test-application procedure is initiated by shifting in the first test pattern into the circuit.

3. The scan-out of the first test response and the scan-in of the next pattern are then carried out simultaneously. This process is repeated until all the test patterns are applied to the CUT, and all test responses are shifted out of the circuit.

4. The scan-out of the final test response terminates the test application process for the CUT.

We next compute the cycle-by-cycle power when response R_i is shifted out and test pattern T_j is shifted in, for a scan chain of length n. Let $TC_k(R_i, T_j)$, $1 \leq k \leq n$, denote the power (number of transitions) for shift cycle k. The overall test power can be represented by the following set $TC(R_i, T_j) = \{TC_1(R_i, T_j), \cdots, TC_n(R_i, T_j), TC_{n+1}(R_i, T_j)\}$. The parameter $TC_{n+1}(R_i, T_j)$ denotes the number of transitions during the capture cycle. The average power consumption for $TC(R_i, T_j)$, $\mu(R_i, T_j)$, is given by: $\frac{\sum_{k=1}^{n+1} TC_k(R_i,T_j)}{n+1}$. The unbiased estimate of statistical variance in test power, $\sigma^2(R_i, T_j)$, is given by

$$\sigma^2(R_i, T_j) \quad = \frac{1}{n} \sum_{k=1}^{n+1} (TC_k(R_i, T_j) - \mu(R_i, T_j))^2.$$

For the example of Figure 5.1, the average power consumption and the statistical variance in test power are 3.85 and 0.80, respectively. We use the following two measures as metrics to analyze the variation in power consumption.

1. The first measure is the statistical variance in test power consumption. Let T_{tot} be the test time (in clock cycles) needed to apply all the test patterns for the CUT. Let P_{mean} be the mean value of power consumption per clock cycle during test. The variance in test power consumption for the CUT is defined as $\frac{1}{T_{tot}-1} \sum_{i=1}^{T_{tot}} (P_i - P_{mean})^2$. Low variance indicates low (aggregated) deviation in test power from the mean value of power consumption during test. Successful WLTBI requires the minimization of this metric.

2. The total cycle-to-cycle variation in test power consumption is an indicator of the "flatness" of the power profile during test. Large cycle-to-cycle power variations

are undesirable. We therefore quantify the "flatness" of the power profile using a measure T_{th}, obtained by counting the number of clock cycles i for which $\frac{|P_i - P_{i+1}|}{P_i}$ exceeds a threshold γ. The parameter P_i, denotes the power consumption during the i^{th} clock cycle. A large value of T_{th} for a given value of γ is undesirable for WLTBI.

The optimization problem $\mathcal{P}_{TPO}$ can now be formally stated as follows:

$\mathcal{P}_{TPO}$: Given a CUT with test set $T = \{T_1, T_2, \ldots, T_N\}$, determine an optimal ordering of test patterns such that: 1) the overall variation in power consumption during test is minimized, and 2) the constraint on peak power consumption P_{max} during test is satisfied. As a pre-processing step, the cycle-accurate power information for all pairs of patterns and $\sigma^2(R_i, T_j)$, $\forall i, j$, need to be computed. For N scan chains, each of length n, this step takes $O(nN^2)$ time.

A binary indicator variable x_{ij}, $S \leq i \leq N$, $1 \leq j \leq E$, is used in the optimization problem to ensure that each test pattern appears exactly once in the ordered sequence. It is defined as follows:

$$x_{ij} = \begin{cases} 1 & \text{if } T_j \text{ immediately follows } T_i \\ 0 & \text{otherwise} \end{cases}$$

We use S to denote a (dummy) start pattern and $x_{iS} = 0\ \forall i \Rightarrow 1 \leq i \leq N$. Likewise, E denotes a (dummy) end pattern and $x_{Ei} = 0\ \forall i \Rightarrow 1 \leq i \leq N$.

The objective function for the optimization problem can be written as follows:

$$\text{Minimize } \mathcal{F} = \max_{\forall i} \left\{ \sum_{j=1}^{N} x_{ij} \cdot \sigma^2(R_i, T_j) \right\}$$

The above min-max objective function can be linearized as follows:

$$\text{Minimize } \mathcal{C}, \text{ subject to}$$

$$\mathcal{C} \geq \sum_{j=1}^{N} x_{ij} \cdot \sigma^2(R_i, T_j), 1 \leq i \leq N$$

Next we formulate constraints to ensure that a test pattern is followed (and preceded) by exactly one pattern. This constraint can be represented by the following two sets of equations.

$$\sum_{j=1}^{N} x_{ij} = 1, i = S, 1, 2, \ldots, N$$

$$\sum_{i=1}^{N} x_{ij} = 1, j = 1, 2, \ldots, E$$

We next formulate constraints imposed by the upper limit on peak power consumption during any given clock cycle. Let us assume that the maximum constraint on peak power consumption at any given clock cycle is P_{max}; the constraint to ensure that this limit on power consumption is never violated can be written as:

$$x_{ij} = 0 \text{ if } \max\{TC(R_i, T_j)\} > P_{max}$$

Thus far, the model does not consider the change in power consumption when three test patterns T_i, T_j, T_k are applied consecutively. It is important during WLTBI to ensure that the power consumption between any two consecutive test patterns does not change dramatically. We therefore need to maintain the change in test power between two consecutive patterns within a reasonable threshold TC_{th}. This value is chosen starting with the lowest value of TC_{th} necessary to formulate a valid ordering. We model this constraint as follows:

$$\frac{|TC_n(R_i, T_j) - TC_1(R_j, T_k)|}{TC_n(R_i, T_j)} > TC_{th} \implies x_{ij} \cdot x_{jk} = 0.$$

Minimize $\mathcal{C}$, subject to :

1) $\mathcal{C} \geq \left\{ \sum_{j=1}^{N} x_{ij} \cdot \sigma^2(R_i, T_j) \right\} \quad \forall i$

2) $\sum_{j=1}^{N} x_{ij} = 1, i = S, 1, 2, \ldots, N$

3) $\sum_{i=1}^{N} x_{ij} = 1, j = 1, 2, \ldots, E$

4) $x_{i,S} = 0, \forall i$

5) $x_{E,i} = 0, \forall i$

6) $x_{ij} = 0$ if $\max\{TC(R_i, T_j)\} > P_{max}$

7) $\frac{|TC_n(R_i,T_j) - TC_1(R_j,T_k)|}{TC_n(R_i,T_j)} > TC_{th} \implies u_{ijk} = 0$

8) $x_{ij} + x_{jk} \leq u_{ijk} + 1$

9) $x_{ij} + x_{jk} \geq 2\, u_{ijk}$

/* Constants : TC_{th}, P_{max}, N, $\sigma^2(R_i, T_j)$, $\max\{TC(R_i, T_j)\}$ */

/* Variables : u_{ijk}, x_{ij}, x_{jk} */

Figure 5.2: Integer linear programming model for $\mathcal{P}_{TPO}$.

The $x_{ij} \cdot x_{jk}$ product term is nonlinear and it can be replaced with a new binary variable u_{ijk} and two additional constraints [85]:

$$x_{ij} + x_{jk} \leq u_{ijk} + 1$$

$$x_{ij} + x_{jk} \geq 2\, u_{ijk}$$

In the worst case, the number of variables in the above ILP model is O(N^3) and the number of constraints is also O(N^3). The complete ILP model is shown as Figure 5.2.

5.2.1 Computational complexity of $\mathcal{P}_{TPO}$

It can be easily shown that the pattern-ordering problem for WLTBI is NP-Complete. The objective of $\mathcal{P}_{TPO}$ is to determine an ordering of the N test patterns $\langle O_1, O_2, \cdots, O_N \rangle$ that minimizes $\max\{\sigma^2(R_{O_1}, T_{O_2}), \sigma^2(R_{O_2}, T_{O_3}), \cdots, \sigma^2(R_{O_{N-1}}, T_{O_N})\}$. Before we prove that the pattern-ordering problem for WLTBI is NP-Complete, we introduce the bottleneck traveling salesman problem (BTSP) [119]. Consider a set $\{C_1, C_2, \cdots, C_n\}$ of n cities. The problem of finding a tour that visits each city exactly once and minimizes the total distance traveled is known as TSP. In BTSP, we attempt to find a tour that minimizes the maximum distance traveled between any two adjacent cities in the tour. It has been shown in [119] that BTSP is NP-Complete.

Claim: The pattern-ordering problem $\mathcal{P}_{TPO}$ is NP-Complete.

Proof: We know that pattern-ordering problem is in NP because we can verify any solution in polynomial time with a simple $O(N^2)$ examination of all possible pattern combinations for ordering at each instant.

Let $G = (V, E)$ be a complete graph, where $V = \{C_1, \cdots, C_n\}$ is the set of vertices and $E = \{(C_i, C_j) : C_i \neq C_j\}$ is the set of edges. Every edge (C_i, C_j) has an associated weight $w(i, j)$. In the BTSP context, a vertex can be interpreted as a city and the edge weight can be the distance between the cities or the time of travel between the two cities. With these notations, the BTSP problem is to find a tour that minimizes the maximum distance between any two cities in the tour.

The notations for the same graph $G = (V, E)$ can be written in the context of the pattern-ordering problem. In the context of the pattern-ordering problem, a vertex can be interpreted as a test pattern and the edge weight $w(i, j)$ can be used to represent $\sigma^2(R_i, T_j)$, i.e., variation in test power when test response i is scanned out while scanning in test pattern j.

An optimal ordering of test patterns is one that minimizes the maximum value of $\sigma^2(R_i, T_j)$. This is an exact instance of BTSP. An optimal ordering of test patterns that minimizes the maximum value of variation in test power consumption can be found in polynomial time if and only if a tour that minimizes the maximum distance between all two cities in the tour is found in polynomial time. This proves that $\mathcal{P}_{TPO}$

is NP-hard. Since $\mathcal{P}_{TPO}$ is in NP, we conclude that it is NP-Complete. We next present a heuristic technique to solve $\mathcal{P}_{TPO}$ for large problem instances.

5.3 Heuristic methods for test-pattern ordering

The exact optimization procedure based on ILP is feasible only when the number of patterns is less than an upper limit, which depends on the CPU and the amount of available memory. To handle large problem instances, we present a heuristic approach to determine an ordering of test patterns for WLTBI, given the upper limit P_{max} on peak power consumption. The heuristic method consists of a sequence of four procedures. Its objective is similar to that of the ILP technique, i.e., to minimize the overall variation in power consumption during test. We start by determining cycle-accurate test power information for all pairs of test patterns in O(nN^2) time. We next determine the first pattern to be shifted-in, and then iteratively determine the ordering of patterns such that the variation in test power is minimized. The main steps used in the *Pattern_Order* heuristic, as shown in Figure 5.3, are outlined below:

1. In procedure *Power_Determine*, the cycle-accurate information on test power consumption $TC(R_i, T_j)$ is determined for all possible pairs (R_i, T_j).

2. In procedure *Initial_Assign*, the first test pattern to be shifted-in to the circuit is determined. The pattern T_i that yields the lowest value in test power variance, $\sigma(S, T_i)$, is chosen as the first test pattern to be applied. We ensure that the constraint on peak power consumption P_{max} is not violated when T_i is applied to the CUT. The first pattern T_i that is added to the ordered list of test patterns is referred to as $Init_{pat}$.

3. In procedure *Pat_Order*, the subsequent ordering of patterns is iteratively de-

termined. Once $Init_{pat}$ is determined, the subsequent ordering of patterns are then iteratively determined by choosing the test pattern that results in the lowest test-power variance $\sigma(Init_{pat}, T_i)$ without violating P_{max}.

4. In procedure $Final\_Assign$, the lone unassigned test pattern is added last to the test ordering. A final list of ordered patterns for WLTBI can now be constructed using information from the $Initial\_Assign$ and the Pat_Order procedures.

A search operation is performed each time procedures $Initial\_Assign$, and Pat_Order are executed to determine the test pattern to be ordered. Hence the worst-case computational complexity of the heuristic procedure, not including the $\mathrm{O}(nN^2)$ initialization step, is $\mathrm{O}(N\log_2 N)$.

A second heuristic method based on the ILP model for $\mathcal{P}_{TPO}$ can also be used to determine an ordering of patterns for WLTBI. The computational complexity associated with the ordering of a large number of test patterns limits the use of the ILP model for large circuits. Using a divide-and-conquer approach, the ILP model can recursively be applied to two or more subsets of test patterns for a circuit with large N. The ordered subsets of patterns can then be combined, by placing subsets that result in minimum cycle-to-cycle variation in power consumption adjacent to each other.

5.4 Baseline approaches

In order to establish the effectiveness of the optimization framework for WLTBI, we consider three baseline methods. The first baseline method finds an ordering of test patterns that minimizes the average power consumption during test. The second baseline method finds an ordering of test patterns to minimize the peak power consumption during test. The third baseline randomly orders the test patterns.

Algorithm 1 $Pattern\_Order$: Test-pattern ordering for WLTBI

```
Let $\sigma^2(R_i, T_j)$ be the standard deviation in test power when response $R_i$ is scanned out and $T_j$ is scanned in;
Let $P_{max}$ be the constraint on peak power consumption;
Let $N_{pat}$ be the total number of test patterns to be applied;
/*Procedure $Power\_Determine$*/
for $i = S$ to $N_{pat}$ do
    for $j = 1$ to $E$ do
        determine cycle-by-cycle power when test response $R_j$ is scanned-out, and test pattern $T_i$ is scanned-in;
        determine $\sigma^2(R_j, T_i)$;
        determine $P_{max}(R_j, T_i)$;
    end for
end for
/*Procedure $Initial\_Assign$*/
$Length = N_{pat}$
min$_{var} = \infty$;
for $i = 1$ to $Length$ do
    if ($\sigma^2(S, T_i) <$ min$_{var}$) && (max$\{TC(S, T_i)\} < P_{max}$) then
        min$_{var} = \sigma^2(S, T_i)$;
        $Init_{pat} = i$;
    end if
end for
first pattern to be ordered = $Init_{pat}$;
$Length = N_{pat} - 1$
/*Procedure $Pat\_Order$*/
repeat
    for $i = 1$ to $Length$ do
        if ($\sigma^2(Init_{pat}, Length(i)) <$ min$_{var}$) && (max$\{TC(Init_{pat}, Length(i))\} < P_{max}$) then
            min$_{var} = \sigma^2(Init_{pat}, Length(i))$;
            $Init_{pat} = Unordered(i)$;
        end if
    end for
    $Length = Length - 1$;
    Update list of unassigned patterns;
until $Length = 0$;
/*Procedure $Final\_Assign$*/
Last reordered pattern = $Unordered(i)$;
Construct final ordered list of patterns for WLTBI.
```

Figure 5.3: Pseudocode for the $Pattern\_Order$ heuristic.

5.4.1 Baseline method 1: Average power consumption

The first baseline method determines an ordering of test patterns to minimize the average power consumption during test. The problem of reordering test sets to minimize average power has been addressed using the well-known TSP [67, 68]. Starting with the initial state S, consecutive test patterns are selected at each instance to minimize the average power consumption.

The above problem can be easily shown to be NP-hard [112]. Efficient heuristics are therefore necessary to determine an ordering of test patterns to minimize the average power consumption in a reasonable amount of CPU time. We use a heuristic technique based on the cross-entropy method [120]. The average power values are collected in a matrix of size $N \times N$. Each element in the matrix corresponds to an average power value for an ordered pair of patterns; for example element $(1, 2)$ in the matrix corresponds to the average power consumption when test pattern 2 is shifted-in after test pattern 1. The heuristic technique takes the complete $N \times N$ matrix as an input to determine an ordering of test patterns.

5.4.2 Baseline method 2: Peak power consumption

The second baseline approach determines an ordering of test patterns such that the peak power consumption is minimized during test. The objective function for this baseline method is as follows:

$$\text{Minimize } \mathcal{F} = \max_{\forall i}\left\{ \sum_{j=1}^{N} x_{ij} \cdot \mathcal{P}(R_i, T_j) \right\},$$

where $\mathcal{P}(R_i, T_j)$ denotes the peak power consumption when response R_i is shifted out while simultaneously shifting in T_j. This optimization problem can be easily solved to obtain a test-pattern ordering that reduces the peak power consumption. As in the case of $\mathcal{P}_{TPO}$, an ILP method can be used for this baseline for small problem instances. For large problem sizes, procedures *Initial_Assign* and *Pat_Order* can be modified to select a test-pattern ordering that results in the lowest peak power consumption.

5.5 Experimental results

In this section, we present experimental results for eight circuits from the ISCAS'89 test benchmarks, and five IWLS'05 circuits. Since the objective of the test pattern ordering problem is to minimize the variation in test power consumption during WLTBI, we present the following results:

- The percentage difference in variance between baseline method 1 and the *Pattern _Order* heuristic. This difference is denoted by δV_{B1}, and it is computed as $\frac{V_{Baseline1} - V_{Pattern\_Order}}{V_{Baseline1}} \times 100\%$; $V_{Pattern\_Order}$ represents the variance in test power consumption obtained using the *Pattern_Order* heuristic, and $V_{Baseline1}$ represents the variance in power consumption obtained using the second baseline method.

- The percentage difference in variance between baseline method 2 and the *Pattern _Order* heuristic. This is calculated in a similar fashion as $\delta V_{Baseline1}$, and is denoted as δV_{B2}.

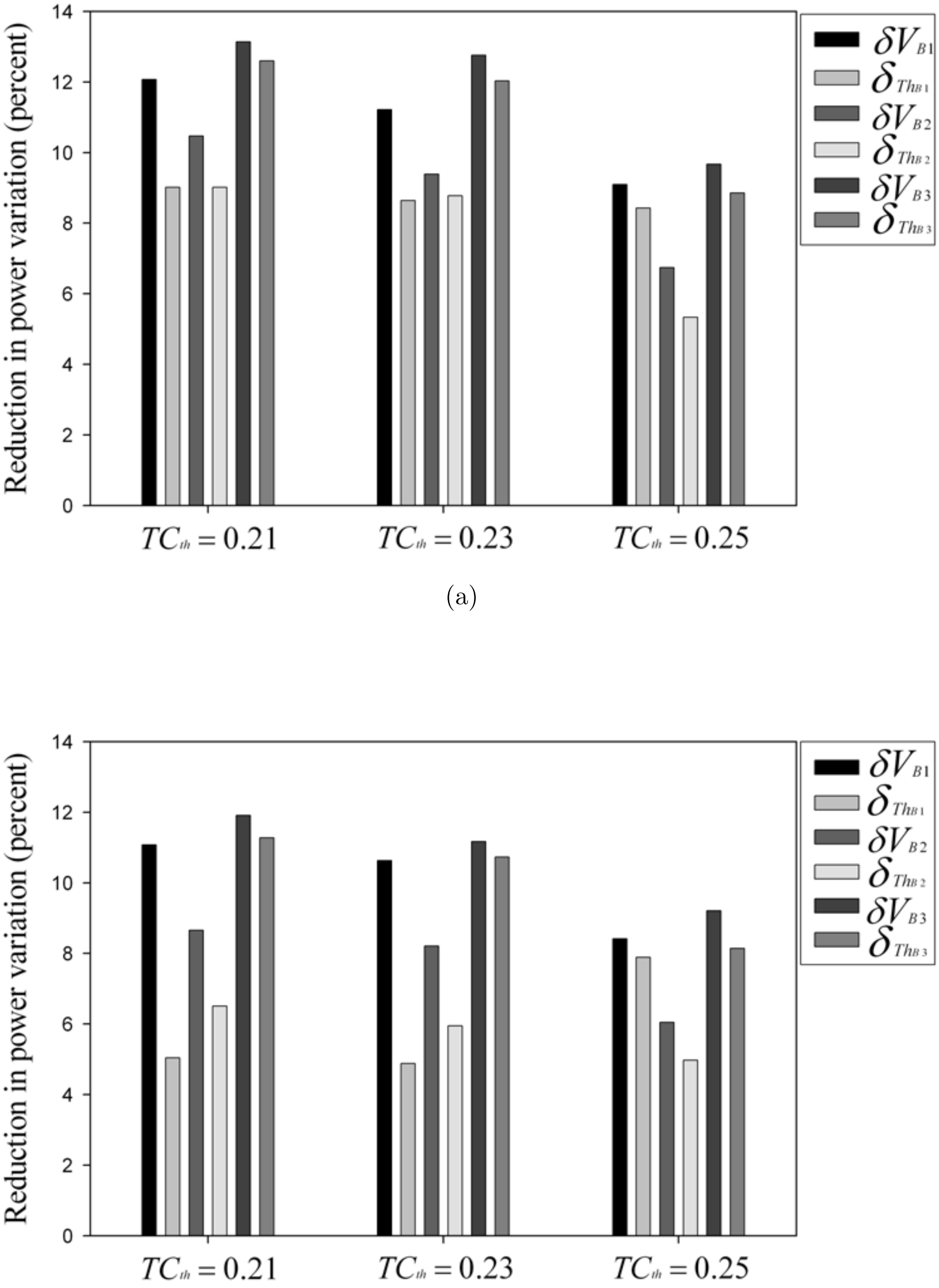
- The percentage difference in variance obtained using random ordering of test patterns and the *Pattern_Order* heuristic. This is calculated in a similar fashion as $\delta V_{Baseline1}$, and is denoted as δV_{B3}.

- We highlight the difference in the total number of clock cycles i during which $\frac{|P_i - P_{i+1}|}{P_i}$ exceeds γ for baseline method 1, and *Pattern_Order*. We characterize this difference as $\delta T_{th_{B1}} = \frac{T_{th_{Baseline1}} - T_{th_{Pattern\_Order}}}{T_{thBaseline1}} \times 100\%$; $T_{th_{Baseline1}}$ and $T_{th_{Pattern\_Order}}$ are the measures (defined in Section IV) obtained using the first baseline method and the *Pattern_Order* heuristic respectively. The value of γ is chosen to be 0.05 (i.e., 5%) to highlight the flatness in power profiles obtained using the different techniques.

- The indicators $\delta T_{th_{B2}}$ and $\delta T_{th_{B3}}$ are determined in a similar fashion as $\delta T_{th_{B1}}$.

Table 5.1: Percentage reduction in the variance of test power consumption obtained using ILP and the *Pattern_Order* heuristic.

				ILP						***Pattern_Order***					
				Baseline 1		**Baseline 2**		**Baseline 3**		**Baseline 1**		**Baseline 2**		**Baseline 3**	
Circuit	N	n	P_{max}	δV_{B1}	$\delta T_{th_{B1}}$	δV_{B2}	$\delta T_{th_{B2}}$	δV_{B3}	$\delta T_{th_{B3}}$	δV_{B1}	$\delta T_{th_{B1}}$	δV_{B2}	$\delta T_{th_{B2}}$	δV_{B3}	$\delta T_{th_{B3}}$
s1423	94	98	60	13.80	13.73	12.39	17.84	12.86	12.13	10.57	11.60	10.03	14.11	11.12	10.64
			65	11.12	11.39	10.21	14.10	10.79	10.48	8.73	9.08	7.91	11.62	7.53	7.06
s5378	155	313	145	12.07	9.02	10.47	9.02	13.14	12.60	8.63	8.91	7.42	6.03	8.15	7.87
			150	11.08	5.04	8.66	6.51	11.91	11.28	7.01	5.00	5.72	5.86	7.56	7.01
s35392	66	2083	1080	10.53	11.57	7.13	6.42	10.88	10.12	6.57	7.36	5.40	6.91	6.07	5.74
			1120	8.64	7.40	6.94	9.15	7.23	6.86	5.32	4.06	4.19	5.11	5.57	5.13
ac97_ctrl	106	2252	1210	9.13	10.32	6.88	7.21	11.12	11.64	7.91	8.10	6.49	7.33	9.87	10.23
			1220	6.93	6.97	6.08	6.37	8.04	8.63	6.15	6.58	5.79	5.90	7.61	8.11
			1230	6.91	6.94	6.07	6.33	8.00	8.58	6.09	6.52	5.71	5.88	7.55	7.97



(a)

(b)

Figure 5.4: Impact of TC_{th} on test power variation for s5378: (a) $P_{max} = 145$ and (b) $P_{max} = 150$.

- For three ISCAS'89 and one IWLS'05 benchmark circuits, the above results are reported for both the ILP method and the *Pattern_Order* heuristic.
- For three benchmark circuits, the above results are reported for the ILP-based heuristic technique.
- The reduction in the variance of test power are reported for three ISCAS'89 benchmark circuits with a single scan chain, using t-detect test sets.

We use a commercial ATPG tool to generate t-detect ($t = 1, 3, 5$) stuck-at patterns (and responses) for the ISCAS'89 and IWLS'05 benchmarks. The experimental results for three ISCAS'89 benchmark circuits obtained using ILP and the *Pattern_Order* heuristic is shown in Table 5.1. Figure 5.4 illustrates the impact of TC_{th} on the percentage savings in test power variation. It is observed that higher (relaxed) values of TC_{th} result in reduced savings in test power variation for s5378; similar results are observed for other circuits. The results for five larger ISCAS'89 benchmark circuits are listed in Table 5.2. The experimental results for the five IWLS'05 benchmark circuits are listed in Table 5.3. The values of P_{max} (measured in terms of the number of flip-flop transitions) for each circuit are chosen carefully after analyzing the per-cycle test-power data. We also present experimental results obtained using the ILP-based heuristic technique for three benchmark circuits in Table 5.4. We use the smallest value of TC_{th} necessary to construct a valid ordering for the results in Table 5.4. Experimental results obtained using t-detect test sets for three ISCAS'89 circuits are presented in Table 5.5.

The ordering of test patterns using the ILP based technique yields lower variation in test power compared to the heuristic method. The *Pattern_Order* heuristic however, is an efficient method for circuits with a large number of test patterns. The results show that a significant reduction in test power variation can be obtained using the proposed ordering technique. The test-pattern-ordering technique also results in low cycle-to-cycle variation in test power consumption. The ILP-based heuristic technique can also be used as an effective technique to determine the ordering of test patterns for WLTBI. This reduction in test power variation obtained using the ILP-based heuristic technique is comparable to the *Pattern_Order* heuristic.

Even small reductions in the variations in test power can contribute significantly towards reducing yield loss and test escape during WLTBI. We know from Equation (1.1) that the junction temperature of the device varies directly with the power consumption. This indicates that a 10% variation in device power consumption will lead to a 10% variation in junction temperatures; this can potentially result in thermal runaway (yield loss), or under burn-in (test escape) of the device. The importance of controlling the junction temperature for the device to minimize post-burn-in yield loss is highlighted in [40].

All experiments were performed on a 2.4 GHz AMD Opteron processor, with 4 GB of memory. The CPU times for optimal ordering of test patterns using ILP ranges from 16 minutes for s1423 to 6 hours for s5378. The CPU times for ordering test patterns using the *Pattern_Order* heuristic, when the cycle-accurate power information is given, is in the order of minutes (the maximum being 120 minutes for s13207). The CPU time to construct the cycle-accurate power information is in the order of hours for the benchmark circuits.

5.6 Summary

We have formulated a test-pattern-ordering problem to minimize power variations during WLTBI. The pattern-ordering approach is based on cycle-accurate power information for the device under test. An exact solution technique has been developed based on integer linear programming. Heuristic techniques have also been presented

Table 5.2: Percentage reduction in the variance of test power consumption obtained using the *Pattern_Order* heuristic for selected ISCAS'89 benchmark circuits.

Circuit	$\boldsymbol{N}$	$\boldsymbol{n}$	**No. of scan chains**	$\boldsymbol{P_{max}}$	**Baseline 1**		**Baseline 2**		**Baseline 3**	
					$\boldsymbol{\delta V_{B1}}$	$\boldsymbol{\delta T_{th_{B1}}}$	$\boldsymbol{\delta V_{B2}}$	$\boldsymbol{\delta T_{th_{B2}}}$	$\boldsymbol{\delta V_{B3}}$	$\boldsymbol{\delta T_{th_{B3}}}$
s9234	231	290	1	155	18.50	18.61	12.49	14.13	18.51	19.43
				165	14.16	16.42	10.46	11.21	11.13	12.62
				175	9.54	18.83	6.68	7.57	5.42	5.91
			4	155	16.98	14.39	10.97	9.68	8.62	9.51
				165	7.66	11.13	7.58	8.72	6.33	7.14
				175	5.14	8.92	4.41	5.19	4.02	4.39
			8	155	10.92	13.33	2.60	2.93	7.90	8.17
				165	4.83	9.69	0.91	1.44	4.52	4.75
				175	3.49	6.87	0.41	1.02	3.66	4.13
s13207	311	723	1	460	4.43	5.11	1.12	4.00	4.59	4.72
				470	2.90	3.58	0.97	1.88	3.12	3.37
				480	2.89	3.58	0.97	1.88	3.12	3.37
			4	460	3.56	3.94	0.78	3.24	3.73	4.02
				470	2.19	2.74	0.41	0.61	2.53	2.81
				480	2.19	2.73	0.41	0.61	2.53	2.81
			8	470	1.81	1.62	0.26	1.31	1.99	2.11
				480	1.81	1.62	0.26	1.31	1.99	2.11
s15850	210	761	1	400	16.66	25.71	10.57	14.33	14.71	17.54
				410	11.19	19.19	6.96	8.05	9.42	11.17
				420	8.42	16.11	3.93	3.96	7.11	8.30
			4	410	8.22	14.34	4.95	5.19	6.31	7.02
				420	4.94	9.13	0.14	0.09	5.16	5.93
			8	410	6.33	10.23	3.22	3.17	4.88	5.26
				420	3.75	7.81	≈ 0	0.03	3.64	4.00
s38417	198	764	1	390	4.08	6.39	3.39	3.62	2.59	2.82
				405	3.48	3.56	2.44	2.11	1.67	1.79
				415	0.77	0.81	0.25	0.40	0.08	0.15
			4	405	3.06	3.42	2.16	3.53	1.80	1.94
				415	0.54	0.67	0.09	0.28	≈ 0	0.06
s38584	162	1372	1	695	7.11	4.13	5.94	3.26	8.39	7.44
				710	5.76	3.32	4.19	3.55	6.08	5.64
				720	3.51	2.86	2.84	1.92	4.70	3.98
			4	695	5.83	5.01	4.64	3.91	6.14	5.72
				710	4.46	3.59	3.63	3.04	5.22	4.60
				720	3.49	2.88	2.21	1.64	3.98	3.52
			8	695	3.21	2.61	2.52	2.03	3.68	3.17
				710	2.20	1.75	1.43	1.08	2.76	2.21

Table 5.3: Percentage reduction in the variance of test power consumption obtained using the *Pattern_Order* heuristic for selected IWLS'05 benchmark circuits.

Circuit	N	n	**No. of scan chains**	P_{max}	**Baseline 1**		**Baseline 2**		**Baseline 3**	
					δV_{B1}	$\delta T_{th_{B1}}$	δV_{B2}	$\delta T_{th_{B2}}$	δV_{B3}	$\delta T_{th_{B3}}$
systemcaes	294	1008	1	570	9.55	8.93	7.31	7.08	9.94	9.62
				580	6.67	6.48	4.70	4.58	7.29	7.66
				590	6.53	2.81	4.64	1.90	7.19	7.61
usb_funct	237	1918	1	1030	7.14	6.83	5.91	5.77	7.62	7.48
				1040	4.52	4.04	2.23	1.91	5.05	4.96
				1060	4.27	3.95	1.69	1.34	4.87	4.54
ac97_ctrl	230	2302	1	1055	12.32	11.98	10.61	10.49	12.73	12.24
				1065	12.18	11.43	9.87	9.14	12.45	12.09
				1075	11.66	10.93	9.21	8.87	12.14	11.58
wb_conmax	413	3316	1	1520	5.12	5.08	4.37	4.11	5.91	5.75
				1530	4.63	4.42	4.04	3.98	4.80	4.66
des_perf	346	9105	1	5660	8.39	7.87	6.71	6.63	8.58	8.33
				5670	8.12	7.94	6.27	6.01	7.93	7.67
				5680	7.48	7.20	5.51	5.63	7.62	7.56

Table 5.4: Percentage reduction in the variance of test power consumption obtained using the ILP-based heuristic.

Circuit	$\boldsymbol{N}$	$\boldsymbol{n}$	**No. of scan chains**	$\boldsymbol{P_{max}}$	**Baseline 1**		**Baseline 2**		**Baseline 3**	
					$\boldsymbol{\delta V_{B1}}$	$\boldsymbol{\delta T_{th_{B1}}}$	$\boldsymbol{\delta V_{B2}}$	$\boldsymbol{\delta T_{th_{B2}}}$	$\boldsymbol{\delta V_{B3}}$	$\boldsymbol{\delta T_{th_{B3}}}$
s9234	231	290	1	155	13.94	14.06	8.19	11.76	15.61	17.12
				165	10.39	14.66	7.82	9.24	8.68	9.62
				175	9.54	18.83	6.68	7.57	5.42	5.91
			4	155	14.53	12.14	9.28	10.12	7.43	8.15
				165	8.12	12.41	8.92	9.48	7.74	9.02
				175	4.83	8.24	4.19	4.54	3.67	3.98
			8	155	11.44	14.16	4.43	3.08	8.41	8.93
				165	4.21	8.86	0.63	0.72	3.67	4.06
				175	3.75	7.21	0.92	1.36	4.88	5.61
s13207	311	723	1	460	5.39	6.41	1.46	4.84	5.26	5.97
				470	2.74	3.18	0.93	1.76	2.98	3.05
				480	2.73	3.16	0.92	1.76	2.97	3.03
			4	460	2.90	3.43	0.50	2.52	3.17	3.83
				470	3.18	3.66	0.84	0.92	3.72	4.04
				480	3.14	3.53	0.80	0.89	3.41	3.63
systemcaes	294	1008	1	570	7.82	7.43	5.92	5.84	8.39	8.11
				580	6.33	6.21	4.52	4.17	7.04	6.90
				590	5.86	5.94	4.91	5.14	6.28	6.63

Table 5.5: Percentage reduction in the variance of test power consumption obtained using the *Pattern_Order* heuristic for three ISCAS'89 benchmark circuits using t-detect test patterns.

Circuit	$\boldsymbol{n}$	$\boldsymbol{N}$	$\boldsymbol{t}$**-detect**	$\boldsymbol{P_{max}}$	**Baseline 1**		**Baseline 2**		**Baseline 3**	
					$\boldsymbol{\delta V_{B1}}$	$\boldsymbol{\delta T_{th_{B1}}}$	$\boldsymbol{\delta V_{B2}}$	$\boldsymbol{\delta T_{th_{B2}}}$	$\boldsymbol{\delta V_{B3}}$	$\boldsymbol{\delta T_{th_{B3}}}$
s5378	313	363	$t = 3$	150	8.19	7.63	5.48	5.06	9.26	9.45
				155	6.92	7.14	4.03	3.70	7.57	7.72
		586	$t = 5$	150	7.31	7.65	4.98	5.13	8.10	8.22
				155	7.16	7.49	4.41	4.64	7.73	7.62
s9234	290	349	$t = 3$	150	17.12	19.46	7.03	9.64	13.39	15.94
				160	7.40	9.14	4.32	7.26	9.13	10.29
				170	4.95	5.43	3.89	3.96	7.14	8.41
		539	$t = 5$	150	13.81	15.02	10.48	11.37	16.17	17.26
				160	7.08	7.93	6.42	7.21	9.15	9.87
				170	4.11	5.03	3.74	4.36	5.69	6.56
s38417	764	436	$t = 3$	390	5.14	5.89	3.96	4.08	5.42	5.63
				405	4.78	4.92	3.12	3.33	5.31	5.74
				415	2.01	2.26	0.79	1.07	2.45	2.63
		679	$t = 5$	390	6.38	6.72	5.15	5.41	6.93	6.86
				405	5.85	6.11	4.32	4.47	6.74	6.88
				415	3.53	3.38	2.06	2.23	3.87	4.10

to solve the pattern-ordering problem. We have compared the proposed reordering techniques to baseline methods that minimize peak power and average power, as well as a random-ordering method. In addition to computing the statistical variance of the test power, we have also quantified the flatness of the power profile during test application. Experimental results for the ISCAS'89 and the IWLS'05 benchmark circuits show that there is a moderate reduction in power variation if patterns are carefully ordered using the proposed techniques. Since the junction temperatures in the device under test are directly proportional to the power consumption, even small reductions in the power variance offer significant benefits for WLTBI.

In Chapter 6, we present a unified test-pattern manipulation and pattern-ordering framework for WLTBI. The presence of don't care bits in test cubes is exploited in the next chapter to reduce the variation in power consumption during scan shift and capture.

Chapter 6

Wafer-Level Test During Burn-In (Part 3): Power-Management Framework

In Chapter 5, a test-pattern ordering technique was proposed for WLTBI. It determines an ordering of test patterns for WLTBI while minimizing the variation in power consumption. It was however assumed that the test patterns do not contain any don't-care bits.

Dynamic burn-in using a full-scan circuit ATPG was proposed in [121] with the objective of maximizing the number of transitions in the scan chains. We focus on a WLTBI-specific X-fill framework that can control the variation in power consumption during scan shift/capture. The test-pattern-ordering technique developed in Chapter 5 is integrated into this framework to further reduce the variation in power consumption during WLTBI.

We show how test-data manipulation and pattern ordering can be used to alleviate thermal problems during WLTBI. We present a unified framework for test-pattern-manipulation and test-pattern-ordering for scan-based WLTBI. Our goal is to minimize the variation in test power during test application. In order to fully realize the benefits of WLTBI, it is necessary to address the challenges of test during burn-in at the wafer level. We attempt to reduce the variation in power consumption during test by manipulating test cubes. Improving power-management for WLTBI can result in reduced yield loss at the wafer level [122].

The remainder of this chapter is organized as follows. Section 6.1 provides a description of the metrics used along with a description of the problem. Section 6.2 presents the “minimum variance” framework to control power variation for WLTBI.

The baseline methods used to evaluate the proposed technique are presented in Section 6.3. Section 6.4 presents simulation results for several ISCAS'89 and IWLS'05 benchmark circuits [113]. Finally, Section 6.5 summarizes the chapter.

6.1 Minimum-variation X-fill problem: $\mathcal{P}_{MVF}$

In this section, we present an outline of a procedure that can be used to manage test power efficiently for WLTBI. Test application for a DUT is carried out by simultaneous scan-out of the test response and scan-in of the next test pattern; this is repeated until all the test patterns are applied to the DUT. Every time a shift operation is performed there is significant switching activity in the scan chains. This leads to constantly varying device power during test. It is therefore important to minimize the cycle-by-cycle variation in the number of transitions during the course of pattern application. In addition, it is also important to minimize the power variance for scan capture. The capturing of output responses in the scan chains can result in excessive flip-flop transitions, resulting in the violation of peak-power constraints [71]. In [63], a metric known as the weighted transition count (WTC) was presented to calculate the number of transitions in the scan chain during scan shifting. It was also shown in [63] that the WTC has a strong correlation with the total device consumption. The WTC metric can be easily extended to determine the cycle-by-cycle transition counts during pattern application.

6.1.1 Metrics: Variation in power consumption during test

As in Chapter 5, we use the following two measures as metrics to analyze the variation in power consumption.

1. The statistical variance in test power consumption.

2. The total cycle-to-cycle variation in test power consumption used to assess the "flatness" of the power profile during test.

Detailed descriptions of the above two metrics can be found in Chapter 5 (Section 5.1) of this thesis.

6.1.2 Outline of proposed method

The goal of problem $\mathcal{P}_{MVF}$ is to first determine optimal X-fill values for the test cubes using for scan-based testing, and then an ordering of fully specified test vectors such that the overall variation in power consumption during test is minimized. For simplicity of discussion, we assume a single scan chain for test application and N test cubes $T_1, T_2, \cdots, T_N$. The extension of $\mathcal{P}_{MVF}$ to a circuit with multiple scan chains is trivial. The optimization problem $\mathcal{P}_{MVF}$ can now be formally stated as follows:
$\mathcal{P}_{MVF}$: Given a CUT with a set T of N test cubes, i.e., $T = \{T_1, T_2, \ldots, T_N\}$, determine appropriate X-fill values for the unspecified bits in the test cubes, and subsequently determine an optimal ordering of the fully specified test patterns such that: 1) the overall variation in power consumption during test is minimized, and 2) the constraint on per-cycle peak-power consumption P_{max} during test is satisfied.

The steps involved in the proposed Min_Var procedure to minimize power variation are as follows:

Step 1: The first step involves generation of test cubes for the DUT for any targeted fault set. In our work we consider test patterns for stuck-at faults. An *a priori* random ordering of test cubes for the DUT is first considered.

Step 2: The second step involves the elimination of power violations that occur during scan shifting. The objective during this step is to fill the unspecified bits in the test cube (X-fill) such that the cycle-by-cycle variation in power consumption is minimized. There is significant variation in power consumption when a test response is shifted out and a test pattern is shifted in simultaneously. This procedure minimizes the cycle-by-cycle variation in test power for the pattern ordering determined in the first step.

Step 3: In this step, peak-power violations due to scan capture are eliminated. If capture-power violations are observed after the X-fill procedure, the previously assigned values of Xs are reassigned to new values to control the capture power during test.

Step 4: The penultimate step in the power-management procedure for WLTBI involves test-pattern ordering. After Steps 1, 2 and 3 are completed, the test-pattern ordering approach from [123] is used to further reduce the variation in power consumption during WLTBI.

Step 5: The final procedure checks for any power violations introduced by the test-pattern ordering procedure. Power violations, if any, are resolved in a similar fashion as done in Step 3.

6.2 Framework to control power variation for WLTBI

In this section, we describe Min_Var, a procedural framework to control the power variation during test for WLTBI. It consists of a sequence of four steps as described in Section 6.1.

6.2.1 Minimum-variation X-filling

The switching activity in the scan flip-flops during shift in/out result in significant power consumption during test. Let us consider a scan chain of length n that has an initial value $r = (r_1 \; r_2 \; \cdots \; r_n)$; the initial value corresponds to the test response from previous pattern application. Let us consider a test pattern $t = (t_1 \; t_2 \; \cdots \; t_n)$

that is shifted into the scan chain. Figure 6.1 represents the cycle-by-cycle change in states of the scan cells when the test pattern t is scanned in and the test response $r = (r_1\ r_2\ \cdots\ r_n)$ is scanned out. The total number of transitions in the scan chain, i.e., the transition count for the various clock cycles can be represented by the equations described in Figure 6.2. The transition count during any clock cycle j is represented as $TC(j)$; e.g., $TC(1)$ represents the total number of transitions in the scan chain during the first clock cycle.

Clock cycle	FF_1	FF_2	FF_3	FF_4	$\cdots$	FF_{n-1}	FF_n
0	r_1	r_2	r_3	r_4	$\cdots$	r_{n-1}	r_n
1	t_n	r_1	r_2	r_3	$\cdots$	r_{n-2}	r_{n-1}
2	t_{n-1}	t_n	r_1	r_2	$\cdots$	r_{n-3}	r_{n-2}
3	t_{n-2}	t_{n-1}	t_n	r_1	$\cdots$	r_{n-4}	r_{n-3}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n-1$	t_2	t_3	t_4	t_5	$\cdots$	t_n	r_1
n	t_1	t_2	t_3	t_4	$\cdots$	t_{n-1}	t_n

Figure 6.1: State of the flip-flops during scan testing.

$$\begin{aligned}
TC(1) &= (r_1 \oplus t_n) + (r_1 \oplus r_2) + (r_2 \oplus r_3) + \cdots + (r_{n-1} \oplus r_{n-2}) \\
&\quad + (r_{n-1} \oplus r_n) \\
TC(2) &= (t_n \oplus t_{n-1}) + (r_1 \oplus t_n) + (r_2 \oplus r_1) + \cdots + (r_{n-3} \oplus r_{n-2}) \\
&\quad + (r_{n-1} \oplus r_{n-2}) \\
&\ \ \vdots \\
TC(n) &= (t_1 \oplus t_2) + (t_3 \oplus t_2) + (t_4 \oplus t_3) + \cdots + (t_n \oplus t_{n-1}) \\
&\quad + (r_1 \oplus t_n)
\end{aligned}$$

Figure 6.2: Total number of transitions for different clock cycles.

The cycle-by-cycle change in transition counts can now be represented using the equations shown in Figure 6.3. The objective during WLTBI is to minimize the cycle-by-cycle change in power consumption during test. This can be accomplished by minimizing the change in transition counts between any two consecutive clock cycles. In other words, our goal is to minimize $\Delta TC(j) = TC(j) - TC(j-1)$ for all j, $2 \leq j \leq n$, by making it as close to 0 as possible.

$$\boxed{\begin{aligned}
&\mathcal{E}_0: && \Delta TC(1) = TC(1)\\
&\mathcal{E}_1: && \Delta TC(2) = |TC(2) - TC(1)|\\
& && \qquad\qquad = |(t_n \oplus t_{n-1}) - (r_{n-1} \oplus r_n)|\\
&\mathcal{E}_2: && \Delta TC(3) = |TC(3) - TC(2)|\\
& && \qquad\qquad = |(t_{n-1} \oplus t_{n-2}) - (r_{n-1} \oplus r_{n-2})|\\
& && \vdots\\
&\mathcal{E}_{n-1}: && \Delta TC(n) = |(t_1 \oplus t_2) - (r_1 \oplus r_2)|
\end{aligned}}$$

Figure 6.3: Equations describing the per-cycle change in transition counts.

We start by eliminating the equations where there are no unspecified bits. We have a system of n equations and at most n unknowns $(t_1, t_2, \cdots, t_n)$ on the right-hand side describing the change in transition count between clock cycles. If we consider the set of equations $\{\mathcal{E}_1, \mathcal{E}_2, \cdots, \mathcal{E}_n\}$, we are specifically interested in the equations that have at least one unspecified bit. We begin the filling of unspecified bits by first considering an equation with only one unknown variable. The following theorem shows that such an equation always exists.

Theorem 1. *Given the set of equations $\mathcal{E} = \{\mathcal{E}_1, \mathcal{E}_2, \cdots, \mathcal{E}_n\}$ from Figure 6.3, denoting the per-cycle change in transition counts, there exists at least one equation in $\mathcal{E}$ that has only a single unknown variable.*

Proof. We use the method of contradiction. Every equation in the set $\mathcal{E}$ has at most two variables on the right-hand side. Suppose none of these equations has exactly one unknown variable. This implies that every equation has two unknowns, i.e., the complete test pattern $t_1, t_2, \cdots, t_n$ is unspecified. This is a contradiction since the test pattern must have at least one specified bit. □

Once the equation $\mathcal{E}_i$ with exactly one unknown is solved to minimize $\Delta TC(i+1)$, it leads to at least another equation with exactly one unknown variable. This process is continued until all the variables are assigned values to minimize each $\Delta TC(j)$, $1 \leq j \leq n$.

Let us consider Equation (6.1) representing the change in transition count for clock cycle j.

$$\Delta TC(j) = t_{n-j+2} \oplus t_{n-j+1} - r_{n-j+2} \oplus r_{n-j+1} \tag{6.1}$$

Without loss of generality, let us suppose that t_{n-j+1} is a care bit and t_{n-j+2} is an unspecified bit. Since our objective is to minimize $\Delta TC(j)$, we can determine t_{n-j+2} as follows:

$$t_{n-j+2} = (r_{n-j+2} \oplus r_{n-j+1}) \oplus t_{n-j+1} \tag{6.2}$$

Once t_{n-j+2} is determined, we delete the equation for $\Delta TC(j)$ from the set of equations and proceed in a similar fashion until all the unspecified bits in the test cubes are filled. As a final step, we solve for $\Delta TC(1)$. It is important to note here that we cannot guarantee the least possible value for $\Delta TC(1)$. However, the above $O(n)$ algorithm is optimal for $\Delta TC(2), \Delta TC(3), \cdots, \Delta TC(n)$ for minimizing variation in power consumption during scan shift. The algorithm solves one equation at a time, and there can be a maximum of n equations; the complexity of the algorithm is therefore $O(n)$.

Previous test response (r)	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}
	0	1	1	0	1	0	1	1	0	1
Original test cube (t)	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
	0	X	X	1	0	X	1	0	X	1
Fully specified test vector after Min_Var fill	0	1	0	1	0	0	1	0	1	1
Fully specified test vector after adjacent fill [19]	0	0	0	1	0	0	1	0	0	1

Figure 6.4: Example to illustrate minimum-variation X-fill.

We next present an example to illustrate the minimum variance X-fill method. Figure 6.4 considers a test-response r that needs to be shifted out while test-pattern t is shifted in. The test-pattern t has unspecified bits that need to be appropriately filled. The fully specified test pattern obtained using the proposed technique is shown in Figure 6.4; the test vector derived from adjacent fill [64] for peak-power minimization is also shown in the figure. The minimum-variance X-fill method results in an X-filling of the test cube that yields 22.47% less cycle-by-cycle variance in test power when compared with the baseline adjacent-fill method.

6.2.2 Eliminating capture-power violations

Capture-power violations occur when an excessive number of flip-fops transition during scan capture. The Hamming distance between the test pattern and the corresponding response quantifies the capture power in terms of the number of transitions. The capture power for a given set of test cubes can be controlled by reassigning 1/0 values to the unspecified bits in the test cubes. The don't-cares in our framework have thus far been mapped to 0s and 1s based on the shift cycles, as described in Section 6.2.1. Let the response captured be denoted by $r^* = (r_1^*, r_2^*, \cdots, r_n^*)$. The following equation now denotes the number of transitions during the capture cycle.

$$\Delta TC(n+1) = t_1 \oplus r_1^* + t_2 \oplus r_2^* + \cdots + t_n \oplus r_n^* \tag{6.3}$$

A capture-power violation occurs when the value of $\Delta TC(n+1)$ exceeds P_{max}. It is therefore necessary to undo the assignment of some of the don't-cares in the test pattern to obtain a permissible value of power during the capture cycle. If $t_i \neq r_i^*$ and if t_i was originally a don't-care, we can reverse the original mapping by complementing its value. Fault-free simulation is then perform with the modified input vector to analyze the impact on capture power, i.e., a check is performed to observe if $\Delta TC(n+1)$ has decreased. If the bit-reversal results in a decrease in $\Delta TC(n+1)$, the change is kept. This procedure is repeated until the capture power violation is resolved. We begin the procedure by assuming a value of power consumption P_{max}^*. If P_{max} denotes the value of power consumption that resolves all power violations, the number of iterations in the procedure is $\frac{(P_{max} - P_{max}^*)}{\Delta P}$; ΔP is the increment step in maximum power consumption during each iteration. The value of ΔP can be chosen based on the size of the benchmarks circuit and constraints on CPU time.

If we assume that the maximum number of unspecified bits in a test cube is p, the worst-case complexity of this procedure is $O(Np)$. It is important to note here that the above procedure does not explore the exponential number of input assignments (2^p assignments in the worst case) for each test cube. The procedure uses a greedy algorithm to save CPU time. Once the capture power violation is resolved, fault-free simulation is performed to verify if the bit-reversals have created new shift-power violations. If power violations exist after the completion of the bit-reversal procedure, the power constraint P_{max} is relaxed and the procedure is repeated; this process is repeated until all power violations are eliminated.

6.2.3 Test-pattern ordering for WLTBI

In Chapter 5 a heuristic method ($Pattern\_Order$) was presented to order test patterns for WLTBI. We use the same test-pattern ordering method here to further reduce the variation in power consumption during WLTBI. Section 5.5 describes the heuristic method in detail. The heuristic approach determines an ordering of test patterns for WLTBI, given an upper limit P_{max} on peak power consumption. It consists of a sequence of four procedures. The main steps used in the $Pattern\_Order$ heuristic, as described in Chapter 5, are outlined below for the sake of completeness:

1. In procedure $Power\_Determine$, the cycle-accurate information on test power consumption $TC(R_i, T_j)$ is determined for all possible response/test-pattern pairs (R_i, T_j).
2. In procedure $Initial\_Assign$, the first test pattern to be shifted-in to the circuit is determined. The pattern T_i that yields the lowest value in test power variance,

$\sigma(S, T_i)$, is chosen as the first test pattern to be applied; S is used to denote a (dummy) start pattern. We ensure that the constraint on peak power consumption P_{max} is not violated when T_i is applied to the CUT. The first pattern T_i that is added to the ordered list of test patterns is referred to as $Init_{pat}$.

3. In procedure Pat_Order, the subsequent ordering of patterns is iteratively determined. Once $Init_{pat}$ is determined, the subsequent ordering of patterns are then iteratively determined by choosing the test pattern that results in the lowest test-power variance $\sigma(Init_{pat}, T_i)$ without violating P_{max}.

4. In procedure $Final\_Assign$, the lone unassigned test pattern is added last to the test ordering. A final list of ordered patterns for WLTBI can now be constructed using information from the $Initial\_Assign$ and the Pat_Order procedures.

6.2.4 Complete procedure

The complete framework for reducing the variation in power consumption during WLTBI is described in Figure 6.5. The process begins by determining the test cubes for the DUT. Starting with a randomly ordered test set, the procedures described in Sections 6.2.1-3 are performed in the order described in Figure 6.5 to obtain an ordered set of fully-specified test patterns. This pattern set is specifically determined for WLTBI in order to keep the fluctuations in junction temperature under control while applying test patterns during burn-in. The experimental results for our proposed framework are described in Section 6.4, and are compared with appropriate baseline scenarios.

We next present an example using the full-scan version of the s208 ISCAS'89 benchmark circuit to illustrate the complete procedure. This circuit has eight flip-flops in the scan chain. We use a commercial ATPG tool to generate test cubes for s208; we consider six of the test cubes for this example, as shown in Figure 6.6(a).

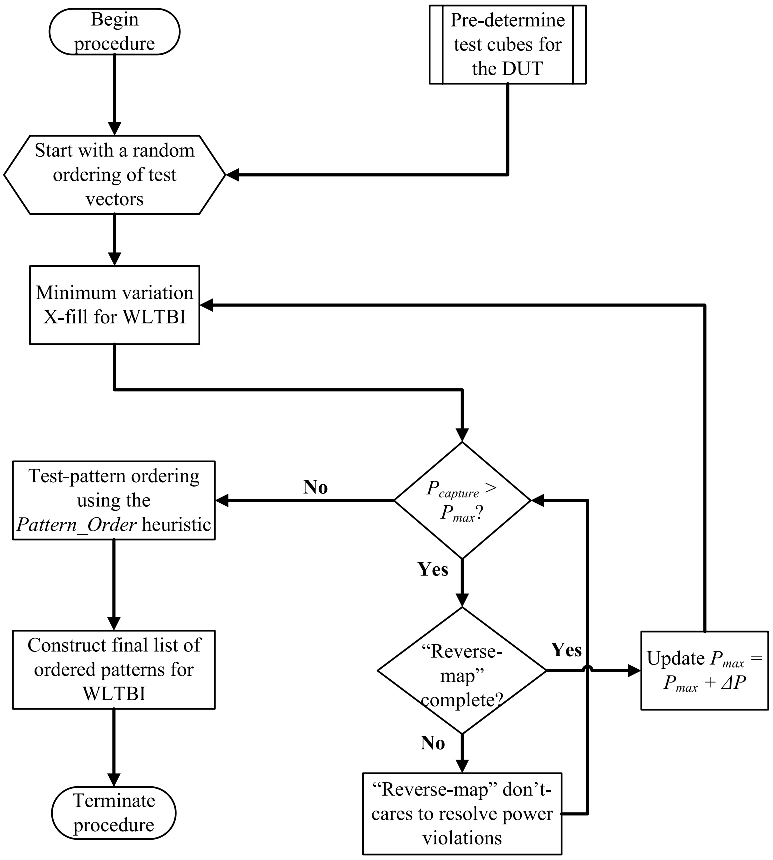


Figure 6.5: Flowchart depicting the Min_Var framework for WLTBI.

The eight equations for the X-fill procedure for the first pattern are shown in Figure 6.6(b). We first target equation $\mathcal{E}_7$, which has one unknown variable t_2. For the first test cube, we set t_2 to 0 to minimize $\Delta TC(8)$. We next consider equation $\mathcal{E}_6$ to minimize $\Delta TC(7)$. This procedure is continued until all X's are assigned values. The same procedure is repeated for the remaining test cubes. The completely-specified test patterns are now shown in Figure 6.6(c). The next step is to check for power violations during capture. We arbitrarily set $P_{max} = 6$ for this example. For the current assignment of don't-care values, a power violation occurs for the current specification of test pattern 3. We use the procedure in Section 6.2.2 to reverse-map don't cares to resolve the power violation. Modifying the third test pattern to $t_3 = 11101010$ removes the power violation. The complete set of previous state test responses are shown in Figure 6.6(c). Finally, we use the procedure in Section 6.2.3 to determine an ordering of test patterns. The ordering of test patterns for this example is $\{3, 2, 6, 1, 4, 5\}$.

Pattern	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
1	1	X	X	X	X	0	X	1
2	X	1	1	X	X	X	1	X
3	X	X	1	0	X	X	X	0
4	0	X	X	X	X	0	1	X
5	1	X	X	0	1	0	X	0
6	X	1	X	X	X	X	0	X

(a)

$$\begin{array}{ll}
\mathcal{E}_0: & \Delta TC(1) = TC(1) \\
\mathcal{E}_1: & \Delta TC(2) = (1 \oplus t_7) - (0 \oplus 0) \\
\mathcal{E}_2: & \Delta TC(3) = (t_7 \oplus 0) - (0 \oplus 0) \\
\mathcal{E}_3: & \Delta TC(4) = (0 \oplus t_5) - (0 \oplus 0) \\
\mathcal{E}_4: & \Delta TC(5) = (t_5 \oplus t_4) - (0 \oplus 0) \\
\mathcal{E}_5: & \Delta TC(6) = (t_4 \oplus t_3) - (0 \oplus 0) \\
\mathcal{E}_6: & \Delta TC(7) = (t_3 \oplus t_2) - (0 \oplus 0) \\
\mathcal{E}_7: & \Delta TC(8) = (t_2 \oplus 1) - (0 \oplus 0)
\end{array}$$

(b)

Pattern	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
1	1	0	0	0	0	0	1	1
2	0	1	1	0	0	1	1	0
3	0	0	1	0	1	0	1	0
4	0	1	1	1	0	0	1	0
5	1	1	0	0	1	0	1	0
6	1	1	1	0	0	1	0	1

(c)

Figure 6.6: (a) Test cubes for s208 benchmark circuit; (b) Equations describing the per-cycle change in transition counts (c) Test set after minimum-variation X-fill.

6.3 Baseline approaches

In order to establish the effectiveness of the proposed framework for WLTBI, we consider four baseline methods. All four baseline methods determine the assignment of the X's in the test cubes to minimize power consumption.

6.3.1 Baseline method 1: Adjacent fill

The first baseline method involves filling strings of X's in the test cube with the same value [124]; this minimizes the number of transitions during scan-in. For example, if we consider a test cube $0X1XXX10$, an assignment of X's that minimizes the number of transitions results in a fully specified test vector 00111110. If a peak-power violation is observed, a reverse bit-stripping process [124] is employed to introduce a different assignment of values to the unspecified bits in the vector. This process is repeated until all peak-power violations are eliminated.

6.3.2 Baseline method 2: 0-fill

The second baseline method employs an X-fill methodology that assigns logic value 0 to all unspecified bits in the test cube. This method was employed in [64] for power minimization during scan testing. In [64], the authors do not consider a specific value of peak power; the objective of the work in [64] is to simply minimize the power consumption during test. For this baseline scenario, we check for power violations during scan-in and capture. If power violations occur after filling the test cubes, we perform reverse bit-stripping to eliminate all peak power violations.

6.3.3 Baseline method 3: 1-fill

The third baseline method is similar to baseline method 2. In this baseline method we assign a logic value 1 to all unspecified bits in the test cube. Power violations are checked and reverse bit-stripping as in Baseline Method 2.

Such a fill technique can still result in a peak-power violation during scan-in and capture. Once X-fill is complete, it is necessary to check if the peak power P_{max} is violated.

6.3.4 Baseline method 4: ATPG-compacted test sets

The final baseline method considers an ATPG-compacted test set, i.e., fully specified test vectors are used. The pattern counts for these test sets are significantly less than for the case where test cubes are used. While the proposed method uses more patterns, it is not a serious concern because burn-in times are relatively high in practice.

- The percentage difference in variance between baseline method 1 and the Min_Var procedure. This difference is denoted by δV_{B1}, and it is computed as $\frac{V_{Baseline1} - V_{Min\_Var}}{V_{Baseline1}} \times 100\%$; V_{Min_Var} represents the variance in test power consumption obtained using the Min_Var procedure, and $V_{Baseline1}$ represents the variance in power consumption obtained using the first baseline method.
- The percentage difference in variance between baseline method 2 and the Min_Var procedure. This is calculated in a similar fashion as $\delta V_{Baseline1}$, and is denoted as δV_{B2}.
- The percentage difference in variance obtained using 1-fill of test cubes and the Min_Var procedure. This is calculated in a similar fashion as $\delta V_{Baseline1}$, and is denoted as δV_{B3}.
- The percentage difference in variance obtained using a fully compacted test set and the Min_Var procedure. This is calculated in a similar fashion as $\delta V_{Baseline1}$, and is denoted as δV_{B4}.
- We highlight the difference in the total number of clock cycles i during which $\frac{|P_i - P_{i+1}|}{P_i}$ exceeds γ for baseline method 1, and Min_Var. We characterize this difference as $\delta T_{th_{B1}} = \frac{T_{th_{Baseline1}} - T_{th_{Min\_Var}}}{T_{thBaseline1}} \times 100\%$; $T_{th_{Baseline1}}$ and $T_{th_{Min\_Var}}$ are the measures obtained using the first baseline method and the Min_Var procedure respectively. The value of γ is chosen to be 0.05 (i.e., 5%) to highlight the flatness in power profiles obtained using the different techniques.

- The indicators $\delta T_{th_{B2}}$, $\delta T_{th_{B3}}$ $\delta T_{th_{B4}}$ and are determined in a similar fashion as $\delta T_{th_{B1}}$.

- The contribution of the pattern-ordering procedure (Section 6.2.3) towards reducing the variance in power consumption. This additional contribution is denoted as $\delta Pattern\_Order$ is computed as $\frac{V_{Min\_Var} - V^{X-fill}_{Min\_Var}}{V_{Min\_Var}}$; $V^{X-fill}_{Min\_Var}$ denotes the variance in test power consumption obtained using the minimum-variation X-fill procedure described in Section 6.2.1-2.

- The contribution of pattern-ordering in reducing the number of clock cycles i during which $\frac{|P_i - P_{i+1}|}{P_i}$ exceeds γ, denoted as $\delta PO_{T_{th}}$.

6.4 Experimental results

In this section, we present experimental results (Tables 6.1-6.5) for circuits from the ISCAS'89 benchmarks and the IWLS'05 benchmarks. Since the objective of the test pattern ordering problem is to minimize the variation in test power consumption during WLTBI, we present the following results:

A commercial tool was used to perform scan insertion for the IWLS benchmark circuits, which are available in Verilog format. We used a commercial ATPG tool to generate stuck-at patterns (and responses) for the full-scan ISCAS'89 and IWLS'05 benchmarks. The results for the five large ISCAS'89 benchmark circuits are listed in Table 6.1. The values of P_{max} (measured in terms of the number of flip-flop transitions per cycle) for each circuit are chosen carefully after analyzing the per-cycle test-power data. We also present experimental results for the IWLS'05 benchmark circuits in Table 6.2. A description of the IWLS'05 benchmarks in terms of the number of scan flip-flops and total number of cells is shown in Table 6.2. Tables 6.3 and 6.4 describe the percentage reduction in the variance of test power consumption using

Table 6.1: Percentage reduction in the variance of test power consumption obtained using the *Min_Var* procedure for the ISCAS'89 benchmark circuits.

Circuit	No. of patterns	No. of flip-flops	No. of scan chains	P_{max}	Baseline 1		Baseline 2		Baseline 3	
					δV_{B1}	$\delta T_{th_{B1}}$	δV_{B2}	$\delta T_{th_{B2}}$	δV_{B3}	$\delta T_{th_{B3}}$
s9234	2005	286	1	150	6.12	5.77	10.16	12.69	9.77	12.14
				160	5.43	5.31	10.02	12.31	8.93	10.21
				170	5.17	5.26	9.67	11.04	7.09	5.91
			4	150	5.94	5.47	9.68	11.41	8.62	11.21
				160	5.16	4.82	8.67	11.13	8.09	10.24
				170	4.33	4.29	7.74	8.45	6.23	9.56
			8	150	3.67	3.18	7.46	8.19	4.92	6.31
				160	3.53	2.84	6.02	6.93	3.56	4.87
				170	1.98	1.77	4.13	5.01	1.45	1.62
s15850	3944	761	1	310	8.17	9.49	14.32	15.94	12.17	12.46
				320	7.73	9.18	13.64	14.86	11.25	11.60
			4	310	6.42	6.79	12.16	12.68	11.92	12.23
				320	6.18	6.41	11.72	11.56	10.98	11.63
				755						
			8	310	5.28	5.13	10.62	11.17	10.86	10.43
				320	4.73	5.02	8.66	9.41	9.84	9.10
s35392	9760	2083	1	895	7.43	7.17	9.84	10.63	8.16	8.49
				910	7.02	6.81	8.92	10.19	8.03	7.61
				930	6.37	6.24	8.65	9.42	7.56	7.33
			4	895	6.29	6.67	9.14	9.76	7.62	8.11
				910	5.83	5.69	7.80	8.23	6.49	7.72
				930	5.58	5.61	7.64	7.92	6.13	7.36
			8	895	4.08	5.40	7.41	7.97	6.86	7.04
				910	2.17	2.62	6.18	7.25	6.51	6.74
				930	0.89	1.20	4.53	5.04	5.19	5.37
s38417	10081	1770	1	770	2.63	3.16	4.41	4.24	5.04	6.19
				780	2.48	2.31	3.72	3.91	4.30	4.46
				790	1.83	1.88	3.15	3.29	3.54	3.61
			4	770	1.17	1.92	2.74	2.32	3.18	3.41
				780	0.64	0.83	2.49	1.87	2.71	2.94
				790	−0.29	−0.07	0.72	1.14	1.53	1.44
			8	770	0.89	1.68	2.13	1.94	2.76	3.13
				780	0.56	1.21	1.63	1.88	2.31	2.58
				790	−0.12	0.05	0.60	0.72	1.79	1.85
s38584	14161	1768	1	735	3.23	4.75	5.14	5.97	5.33	6.19
				745	3.18	4.26	4.84	5.30	4.91	5.47
				755	2.91	2.99	4.62	5.01	4.69	5.05
			4	735	2.70	4.43	4.94	5.12	5.29	5.78
				745	2.17	2.32	3.64	4.48	4.16	4.72
				755	1.67	2.08	3.22	4.58	3.97	4.31
			8	735	1.94	2.67	3.13	4.39	4.61	4.78
				745	0.89	1.23	1.96	2.78	3.21	3.60
				755	−0.22	0.34	1.42	1.73	2.29	2.51

Table 6.2: Percentage reduction in the variance of test power consumption obtained using the *Min_Var* procedure for the IWLS'05 benchmark circuits.

Circuit	No. of patterns	No. of flip-flops	No. of cells	No. of scan chains	P_{max}	Baseline 1		Baseline 2		Baseline 3	
						$\boldsymbol{\delta V_{B1}}$	$\boldsymbol{\delta T_{th_{B1}}}$	$\boldsymbol{\delta V_{B2}}$	$\boldsymbol{\delta T_{th_{B2}}}$	$\boldsymbol{\delta V_{B3}}$	$\boldsymbol{\delta T_{th_{B3}}}$
systemcaes	10454	1058	17817	1	530	8.06	7.92	11.35	11.91	10.62	11.56
					540	7.48	8.16	11.27	11.47	10.26	11.10
					550	6.82	7.33	10.61	11.14	10.83	11.59
				4	530	7.44	7.13	10.87	10.69	10.18	10.96
					540	6.61	7.46	10.54	10.22	9.68	9.70
					550	5.03	5.72	8.61	8.59	7.32	7.93
				8	530	6.17	5.95	8.24	7.98	9.13	9.46
					540	5.82	5.51	8.06	7.62	6.97	7.01
					550	5.37	4.95	7.38	7.14	6.60	6.74
usb_funct	18399	1968	25510	1	960	2.34	3.61	4.46	5.68	4.17	5.36
					970	2.19	3.28	3.67	4.56	3.33	4.18
					980	1.96	2.49	3.12	3.63	3.27	3.78
				4	960	2.17	2.94	4.12	5.51	4.04	4.43
					970	1.89	2.23	3.47	3.82	3.27	3.91
					980	1.56	1.70	2.85	3.29	2.86	3.47
				8	960	1.05	1.46	3.78	4.23	3.59	3.70
					970	0.68	0.87	3.26	3.31	3.04	3.12
					980	0.17	0.23	2.63	2.58	2.52	2.65
ac97_ctrl	20146	2302	28049	1	1025	11.56	12.14	13.12	13.67	12.83	13.58
					1040	11.42	11.53	12.69	12.51	11.40	12.82
					1050	10.93	11.21	11.43	11.60	11.16	11.51
				4	1025	11.13	11.39	12.86	13.23	11.77	12.21
					1040	10.78	10.47	11.94	12.11	10.72	11.63
					1050	10.14	10.39	11.19	11.83	10.38	10.92
				8	1025	10.46	10.24	10.61	11.45	10.18	10.86
					1040	9.80	9.62	10.27	10.95	9.27	10.17
					1050	9.06	8.77	9.83	10.31	8.80	9.51
wb_conmax	57681	3316	59483	1	1460	14.17	15.68	16.33	16.12	17.25	16.98
					1500	13.42	15.03	15.58	15.43	17.02	16.71
				4	1460	13.76	14.47	14.90	15.61	16.36	15.80
					1500	13.19	13.71	14.16	14.87	15.93	15.26
				8	1460	12.93	13.09	14.58	14.12	15.49	15.68
					1500	11.24	10.45	13.31	13.14	14.67	15.11
des_perf	76001	9105	146224	1	5600	7.39	8.12	9.24	10.33	8.75	8.89
					5650	7.14	7.61	9.07	9.96	8.49	8.52
				4	5600	6.73	7.38	8.91	10.14	8.53	8.46
					5650	5.96	7.16	8.12	9.57	7.83	8.13
				8	5600	6.51	6.91	8.43	9.28	7.60	7.71
					5650	5.64	6.83	7.68	8.85	7.27	7.04
ethernet	119636	10752	153945	1	6380	5.17	6.89	8.06	8.17	8.59	8.92
					6400	4.92	5.41	7.73	7.91	8.11	8.44
				4	6380	4.80	4.97	6.76	6.59	7.05	7.14
					6400	4.36	4.48	6.19	6.05	6.52	6.77
				8	6380	4.23	4.47	6.19	6.31	5.95	5.86
					6400	3.95	4.16	5.74	5.86	5.38	5.23

Table 6.3: Percentage reduction in the variance of test power consumption using the Min_Var procedure over Baseline 4 for the ISCAS'89 benchmark circuits.

Circuit	No. of compacted test patterns	No. of scan chains	P_{max}	Baseline 4	
				$\boldsymbol{\delta V_{B4}}$	$\boldsymbol{\delta T_{th_{B4}}}$
s9234	349	1	150	17.46	18.21
			160	15.48	17.65
			170	14.74	15.83
		4	150	16.95	16.37
			160	14.72	15.97
			170	12.35	12.12
		8	150	10.47	11.15
			160	10.08	9.95
			170	5.63	7.18
s15850	210	1	310	19.46	20.03
			320	18.15	18.63
		4	310	15.48	19.67
			320	14.10	18.71
		8	310	13.64	16.76
			320	11.48	12.59
s35392	20146	1	895	13.32	12.53
			910	12.07	11.23
			930	11.70	10.81
		4	895	12.38	11.92
			910	10.51	11.69
			930	10.35	11.38
		8	895	10.03	10.39
			910	8.31	9.96
			930	6.13	7.42
s38417	436	1	770	9.18	8.87
			780	6.53	5.21
			790	5.70	5.38
		4	770	5.18	6.32
			780	3.49	3.67
			790	3.14	3.29
		8	770	4.43	4.86
			780	3.39	4.12
			790	3.24	3.90
s38584	313	1	735	12.01	10.24
			745	11.30	9.93
			755	10.79	9.45
		4	735	11.54	10.82
			745	8.50	8.83
			755	7.51	8.07
		8	735	7.38	8.93
			745	6.70	6.43
			755	3.31	4.39

the Min_Var over Baseline 4 for the ISCAS'89 and the IWLS'05 benchmark circuits respectively. Table 6.5 describes the individual contribution of X-fill and pattern-ordering procedures in reducing the variation in power consumption during test for five benchmarks.

The Min_Var procedure is an efficient method for circuits with a large number

Table 6.4: Percentage reduction in the variance of test power consumption using the *Min_Var* procedure over Baseline 4 for the IWLS'05 benchmark circuits.

Circuit	No. of compacted test patterns	No. of scan chains	P_{max}	Baseline 4	
				$\boldsymbol{\delta V_{B4}}$	$\boldsymbol{\delta T_{th_{B4}}}$
systemcaes	294	1	530	23.24	25.19
			540	21.57	24.43
			550	19.66	21.94
		4	530	17.88	21.34
			540	15.88	20.36
			550	12.09	15.61
		8	530	9.35	14.98
			540	8.82	13.87
			550	8.14	12.46
usb_funct	237	1	960	16.87	15.43
			970	15.79	14.02
			980	14.13	10.64
		4	960	12.62	8.72
			970	10.99	6.61
			980	9.07	5.04
		8	960	6.10	4.33
			970	3.95	2.58
			980	0.99	0.68
ac97_ctrl	230	1	1025	18.68	19.13
			1040	18.45	18.17
			1050	17.66	17.66
		4	1025	17.34	17.38
			1040	16.79	15.98
			1050	15.80	15.85
		8	1025	15.30	15.63
			1040	14.33	14.68
			1050	13.25	13.38
wb_conmax	413	1	1460	21.34	19.87
			1500	20.21	19.05
		4	1460	19.70	18.34
			1500	18.88	17.37
		8	1460	18.51	16.59
			1500	16.09	13.24
des_perf	346	1	5600	12.14	14.86
			5650	11.73	13.93
		4	5600	11.06	13.51
			5650	9.79	13.10
		8	5600	8.89	12.65
			5650	7.70	12.50
ethernet	2110	1	6380	14.96	14.28
			6400	14.24	13.80
		4	6380	13.07	13.24
			6400	12.49	12.76
		8	6380	10.91	11.35
			6400	10.58	10.80

of test patterns. The results show that a significant reduction in test power variation can be obtained using the proposed framework for test data manipulation and test-pattern-ordering. The technique also results in low cycle-to-cycle variation in test power consumption. The "negative" reduction in Table 6.1 in a few cases can be

Table 6.5: Contribution of pattern-ordering in reducing the variation in test power consumption.

Circuit	No. of scan chains	P_{max}	Baseline 1		Baseline 2		Baseline 3	
			$\delta Pattern\_Order$	$\delta PO_{T_{th}}$	$\delta Pattern\_Order$	$\delta PO_{T_{th}}$	$\delta Pattern\_Order$	$\delta PO_{T_{th}}$
s35392	1	895	24.32	25.89	26.37	26.53	16.26	19.84
		910	28.94	24.49	21.39	16.12	16.95	19.66
		930	23.71	28.09	18.87	23.54	25.65	26.71
	4	895	19.52	17.37	20.07	14.60	26.43	21.04
		910	18.40	27.91	18.44	25.98	24.38	22.33
		930	26.57	14.02	18.35	25.92	22.54	21.94
	8	895	15.90	23.52	21.76	18.13	22.66	27.69
		910	22.68	21.23	21.84	18.90	20.07	23.18
		930	24.86	25.30	20.44	18.68	25.66	28.91
s38584	1	735	19.15	2.43	9.30	−4.09	12.26	12.18
		745	6.56	7.38	7.09	7.46	8.43	10.04
		755	6.03	15.07	9.61	1.67	−4.05	14.33
	4	735	10.68	3.20	4.17	12.25	−0.93	3.53
		745	9.78	13.12	14.90	11.26	6.94	18.15
		755	15.53	−0.54	2.95	9.34	14.07	10.58
	8	735	13.57	4.87	16.42	7.51	11.89	15.41
		745	16.66	16.90	3.82	12.16	10.25	10.11
		755	15.17	14.38	6.43	12.66	18.71	−4.25
ac97_ctrl	1	1025	11.17	9.69	12.68	13.27	11.97	18.93
		1040	13.38	17.22	8.76	16.49	13.25	16.96
		1050	8.09	4.00	9.21	11.48	9.77	6.24
	4	1025	16.37	13.26	14.27	10.10	11.88	9.36
		1040	8.56	8.35	16.09	13.50	12.18	5.42
		1050	5.80	17.06	17.84	4.80	9.80	8.54
	8	1025	11.55	16.54	6.68	14.19	8.72	16.10
		1040	5.35	4.84	13.69	17.93	4.68	6.79
		1050	8.71	6.83	5.01	10.79	16.17	12.67
wb_conmax	1	1460	4.00	2.69	−8.82	−3.12	1.50	−2.77
		1500	−1.48	6.59	−7.74	3.82	2.61	−1.10
	4	1460	−8.08	−4.46	−2.32	−2.92	−9.74	3.04
		1500	3.14	−10.43	4.71	−2.41	−10.84	6.94
	8	1460	−3.50	−6.76	6.80	−5.71	1.59	−8.45
		1500	2.01	−4.57	−8.02	6.97	−9.41	4.16
des_perf	1	5600	−5.42	−2.07	−1.85	1.19	−0.35	−3.91
		5650	−4.90	−3.19	−1.54	1.12	−4.90	2.84
	4	5600	−6.26	−0.91	−2.48	−4.31	−2.31	1.60
		5650	0.11	−3.30	−1.07	−4.74	2.17	1.23
	8	5600	1.90	−3.80	−4.07	0.10	−6.73	−6.83
		5650	0.13	−5.09	−2.18	−2.30	−3.47	−1.12

attributed to the heuristic nature of the pattern-ordering procedure. The negative entries in Table 6.5, which implies that pattern reordering is counter-productive can be explained as follows. The test pattern set is split into multiple sets to reorder patterns for large benchmark circuits. This is done to save CPU time for reordering. The $Pattern\_Order$ heuristic appears to be ineffective for these instances. However, X-fill alone results in significant variance reduction in each case.

Even small reductions in the variations in test power can contribute significantly towards reducing yield loss and test escape during WLTBI. We know from Equation (1.1) that the junction temperature of the device varies directly with the power consumption. This indicates that a 10% variation in device power consumption will lead to a 10% variation in junction temperatures; this can potentially result in thermal runaway (yield loss), or under burn-in (test escape) of the device. The importance of controlling the junction temperature for the device to minimize post-burn-in yield loss is highlighted in [40].

All experiments were performed on a 2.4 GHz AMD Opteron processor, with 4 GB of memory. The CPU times for the Min_Var procedure (including X-fill and pattern reordering) is in the order of hours for large benchmark circuits. The CPU time for X-fill alone is in the order of minutes.

6.5 Summary

We have developed a new X-fill method to minimize power variation during WLTBI. This approach is based on cycle-accurate power information for the device under test. For N test patterns, an $O(N)$ procedure has been presented to solve the X-fill problem for scan-shift and capture. We have further reduced the variation in power consumption by reordering the test-pattern set after minimum-variation X-fill. We have compared the proposed reordering techniques to baseline methods that fill unspecified bits in a test cube with the objective of reducing power consumption during scan testing. In addition to computing the statistical variance of the test power, we have also quantified the flatness of the power profile during test application. Experimental results for the ISCAS'89 and the IWLS'05 benchmark circuits show that there is a moderate to significant reduction in power variation if patterns are carefully manipulated and ordered using the proposed framework. Since the junction temperatures in the device under test are directly proportional to the power consumption, even small reductions in the power variance offer significant benefits for WLTBI.

Chapter 7

Conclusions and Future Work

A prerequisite to assembling 3-D ICs and SiP devices, is the ability to manufacture and test KGD solutions in a cost-effective manner. The research reported in this dissertation explores multiple solutions for wafer-level manufacturing test of SoCs and KGDs. The goal of this research is to provide robust and scalable engineering solutions for wafer-level test and optimization techniques for test planning. According to the ITRS [1], each device in the future can be considered to be a SoC or a 3-D IC (or SiP). The need for flexible test solutions to accommodate increasing integration trends has also been emphasized [1]. The high test cost associated with the testing of these devices motivates the need for effective test techniques and test planning at the wafer level. Significant yield improvements early in the product/process development cycle can be achieved by efficient wafer-level test techniques [125].

In this thesis, we have addressed the design of a test infrastructure at the wafer level. Efficient test techniques at wafer level under resource constraints have also been developed. We have developed test-planning approaches for wafer-level test that address test-resource optimization, defect screening, test scheduling, and test-data manipulation for digital and mixed-signal SoCs, as well as for KGDs. This thesis research also focuses on reducing the capital expenditure on ATE at the wafer level by combining the burn-in and test processes.

7.1 Thesis Contributions

Chapter 2 presented a test-length selection problem for wafer-level testing of core-based SoCs. Theoretical foundations and a statistical model were developed, and techniques were presented to determine defect probabilities for the individual cores in an SoC. An ILP model that incorporates defect probabilities to determine the test-lengths for each core in the SoC was also developed with the objective of maximizing defect screening at wafer sort. The ILP approach presented is computationally efficient and takes only a fraction of a second even for the largest SoC test benchmarks from Philips. Experimental results for the ITC'02 SoC test benchmarks have shown that the test-length selection procedure can lead to significant defect-screening at wafer sort. An efficient heuristic method that scales well for larger SoCs was also presented. A test-length selection problem for RPCT of core-based SoCs was also formulated and solved using ILP and heuristic-based techniques.

Chapter 3 presented a wafer-level defect screening technique for core-based mixed-signal SoCs. Correlation-based signature analysis methods were used for defect screening at the wafer-level for analog cores. A cost model was presented to quantify the savings that result from wafer-level testing. An industrial mixed-signal SoC was used to evaluate the proposed wafer-level test method. The proposed method eliminated the need for expensive mixed-signal ATE at wafer sort, reducing test cost and improving tester efficiency.

Chapter 4 presented a test-scheduling problem for WLTBI of core-based SoCs, that minimizes the variation in test power during test application. The test-scheduling method used cycle-accurate test-power data for the cores. A heuristic technique was used to solve the test scheduling problem. Results for the ITC'02 SoC test benchmarks were presented to illustrate the reduction in power variation obtained using the proposed method.

Chapter 5 presented a test-pattern-ordering problem for WLTBI. An efficient heuristic technique was presented to solve the pattern-ordering problem in addition to an ILP based technique. The proposed reordering techniques were compared with appropriate baseline methods. The relevance of the pattern-ordering problem in the context of WLTBI was further emphasized by quantifying the “flatness” in power profile during test application. Experimental results were presented for several ISCAS’89 and IWLS’05 benchmark circuits to show the reduction in power variation obtained using the proposed pattern-ordering techniques.

Chapter 6 presented a new X-fill method to minimize power variation during WLTBI. An efficient $O(N)$ procedure for N test patterns was presented to solve the X-fill problem for scan-shift and capture. The baseline methods considered filled unspecified bits in a test cube with the objective of reducing power consumption during scan testing. The statistical variance of the test power and the flatness of the power profile during test application were quantified for the proposed methods. Experimental results were presented for the ISCAS’89 and the IWLS’05 benchmark circuits. Reduction in power variations obtained by carefully manipulating and ordering test patterns were presented for several benchmark circuits.

7.2 Future work

This thesis has explored a number of wafer-level test solutions that reduce product cost. The focus has been on new test planning methods for digital SoCs, as well as for mixed-signal SoCs and KGDs. As next-generation semiconductor devices become more integrated with multiple functionalities, a number of new test challenges will continue to emerge. We next summarize future research directions. The topics discussed below are aligned with the theme of intelligently performing test at the wafer level to achieve maximum cost benefits.

7.2.1 Integrated test-length and test-pattern selection for core-based SoCs

In this thesis, we have limited ourselves to test-length selection for core-based SoCs under resource constraints of test time and TAM width. The ultimate objective of wafer sort testing is to maximize the detection of faulty die; this maximizes profit margins by lowering packaging costs. The test-length selection framework proposed in this thesis does not address the issue of pattern grading to choose the reduced test pattern set. In [126], "output deviation" was proposed as a coverage-metric and a test-pattern grading method for pattern-reordering. It was also shown in [126] that test sets that are carefully reordered using such a metric as basis can potentially yield "steep" fault coverage curves.

In practice, random ordering of test patterns generated by commercial ATPG tools can be integrated into the framework proposed in this thesis. This can be used to select test-lengths that yield maximum defect screening probability at wafer sort. However, if the same test pattern set generated by commercial ATPG tools are graded and reordered using techniques similar to the ones proposed in [126], defect screening at wafer sort can significantly be enhanced under resource constraints. The pattern reordering step can be used as a pre- or post-processing procedure to the framework proposed in this thesis.

This research direction will provide an intelligent framework for test-pattern selection. The number of test patterns for wafer sort under resource constraints can be determined using the framework presented in Chapter 2; the choice of test patterns however, can be determined using output deviations. This process of test-pattern selection can potentially lead to improved defect screening at the wafer level under resource constraints.

7.2.2 Multiple scan-chain design for WLTBI

Multiple scan chains have been primarily used in DfT architectures to lower test application times. With increasing emphasis on power consumption, several design techniques for multiple scan designs have been developed [127, 128]. In [127] a single scan chain is partitioned into multiple smaller scan chains to minimize the number of transitions in the scan chains. Thus far, the layout information, information on clock domains, and other geometric constraints have not been incorporated in such design techniques.

WLTBI is an enabling technology for cost-efficient manufacture of next generation KGDs. It is important that the on-die variation in junction temperature is kept to a minimum during WLTBI. Efficient DfT architectures that incorporate multiple scan designs while considering the layout of the scan chains need to be developed for future IC designs. Such architectures will minimize the variation in junction temperature during test application. There are several challenges associated with the development of such techniques:

- Full-chip thermal modeling: Complete thermal modeling is necessary to determine the impact of scan-chain placement on overall device temperature. Commercial finite-element analysis tools such as Flotherm [129] can be used to model the impact of scan chain placement. Circuit simulation, in addition to thermal analysis using commercial tools, is essential to optimally determine the best scan-cell placement for optimal thermal performance during WLTBI.

- Impact of routing: It is also important to consider routing constraints during the design of multiple scan chains [130]. The best scan design for thermal performance during WLTBI will not be the best design in terms of scan-chain routing. It is therefore important to incorporate routing constraints for scan design in the overall framework.

7.2.3 Layout-aware SoC test scheduling for WLTBI

One of the primary benefits of a test scheduling method, specifically suited for WLTBI, is the reduction in power variation during test application. In this thesis, we presented an efficient test scheduling approach to minimize the power variation during test. However, the proposed test scheduling method does not consider the placement of the cores in SoC while formulating the test schedule. The simultaneous testing of cores that are placed close to one another can lead to hot-spots during test application. It is also beneficial to stress the DUT uniformly during WLTBI.

It is therefore important to study the activity patterns of the cores during WLTBI, and construct test schedules accordingly, to flatten the temperature profile of the SoC. Modifying existing academic tools such as HOTSPOT [131], or using commercial tools such as FLOTHERM [129] to incorporate die cooling capabilities under burn-in conditions, and constantly varying device power, will lead to more accurate thermal predictions for WLTBI. This can potentially lead to the minimization of yield loss and test escapes during WLTBI.

Bibliography

[1] International Technology Roadmap for Semiconductors: Assembly and Packaging, http://www.itrs.net/Links/2005ITRS/AP2005.pdf 2005.

[2] M. Bushnell and V. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits.* Kluwer, 2000.

[3] D. Appello, P. Bernardi, M. Grosso, and M. S. Reorda, "System-in-package testing: problems and solutions," *IEEE Design & Test of Computers*, vol. 23, pp. 203–211, May. 2006.

[4] R. K. Gupta and Y. Zorian, "Introducing core-based system design," *IEEE Design & Test*, vol. 14, pp. 15–25, Oct. 1997.

[5] S. Steps, "Cost comparison of wafer-level versus singulated die burn-in & test," in 8th *Annual KGD Workshop*, 2001, http://www.napakgd.com/previous/kgd2001/pdf/3–3_Steps.pdf.

[6] "A comparison of wafer level burn-in & test platforms for device qualification and known good die (KGD) production", http://www.delta-v.com/images/White_Paper_-_Comparing_WLBT_Platforms.pdf.

[7] "IEEE standard testability method for embedded core-based integrated circuits," *IEEE Std 1500-2005*, pp. 1–117, 2005.

[8] V. Iyengar, K. Chakrabarty, and E. Marinissen, "Test wrapper and test access mechanism co-optimization for system-on-chip," *Journal of Electronic Testing: Theory and Applications*, vol. 18, pp. 213–230, Apr. 2002.

[9] A. Sehgal, S. K. Goel, E. J. Marinissen, and K. Chakrabarty, "P1500-compliant test wrapper design for hierarchical cores," in *Proceedings of International Test Conference*, 2004, pp. 1203–1212.

[10] S. Koranne, "A novel reconfigurable wrapper for testing of embedded core-based SOCs and its associated scheduling algorithm," *Journal of Electronic Testing: Theory and Applications*, vol. 18, pp. 415–434, Aug. 2002.

[11] S. K. Goel and E. J. Marinissen, "Effective and efficient test architecture design for SOCs," in *Proceedings of International Test Conference*, 2002, pp. 529–538.

[12] Q. Xu and N. Nicolici, "Modular SoC testing with reduced wrapper count," *IEEE Transactions on Computer-Aided Design*, vol. 24, pp. 1894–1908, Dec. 2005.

[13] K. Chakrabarty, "Optimal test access architectures for system-on-a-chip," *ACM Transactions on Design Automation of Electronic Systems*, vol. 6, pp. 26–49, Jan. 2001.

[14] V. Iyengar, K. Chakrabarty, and E. Marinissen, "Test access mechanism optimization, test scheduling and tester data volume reduction for system-on-chip," *IEEE Transactions on Computers*, vol. 52, pp. 1619–1632, Dec. 2003.

[15] Y. Zorian, E. Marinissen, and S. Dey, "Testing embedded core-based system chips," *IEEE Computer*, vol. 32, pp. 52–60, Jun. 1999.

[16] C. P. Su and C. W. Wu, "A graph-based approach to power-constrained SoC test scheduling," *Journal of Electronic Testing: Theory and Applications*, vol. 19, pp. 45–60, Feb. 2004.

[17] V. Iyengar and K. Chakrabarty, "System-on-a-chip test scheduling with precedence relationships, preemption, and power constraints," *IEEE Transactions on Computer-Aided Design*, vol. 21, pp. 1088–1094, Sep. 2002.

[18] E. Larsson, K. Arvidsson, H. Fujiwara and Z. Peng, "Efficient test solutions for core-based designs," *IEEE Transactions on Computer-Aided Design*, vol. 23, pp. 758–775, May 2004.

[19] W. Zou, S. M. Reddy, and I. Pomeranz, "SoC test scheduling using simulated annealing," in *Proceedings of VLSI Test Symposium*, 2003, pp. 325–330.

[20] L. Yan and J. R. English, "Economic cost modeling of environmental-stress-screening and burn-in," *IEEE Transactions on Reliability*, vol. 46, pp. 275–282, Jun. 1997.

[21] P. C. Maxwell, "Wafer-package test mix for optimal defect detection and test time savings," *IEEE Design & Test of Computers*, vol. 20, pp. 84–89, Sep. 2003.

[22] M. F. Zakaria, Z. A. Kassim, M. P. Ooi, and S. Demidenko, "Reducing burn-in time through high-voltage stress test and Weibull statistical analysis," *IEEE Design & Test of Computers*, vol. 23, pp. 88–98, Sep. 2006.

[23] T. J. Powell, J. Pair, M. John, and D. Counce, "Delta IDDQ for testing reliability," in *Proceedings of VLSI Test Symposium*, 2000, pp. 439–443.

[24] I. Y. Khandros and D. V. Pedersen, *Wafer-level burn-in and test.* U. S. Patent Office, May 2000, Patent number 6,064,213.

[25] T. Mckenzie, W. Ballouli, and J. Stroupe, "Motorola wafer level burn-in and test," in *Burn-in and Test Socket Workshop*, 2001, http://www.swtest.org/swtw_library/2002proc/PDF/T02_Mckenzie.pdf.

[26] P. Pochmuller, *Configuration for carrying out burn-in processing operations of semiconductor devices at wafer level.* U. S. Patent Office, Mar 2003, Patent number 6,535,009.

[27] S. Bhattacharya and A. Chatterjee, “Optimized wafer-probe and assembled package test design for analog circuits,” *ACM Transactions on Design Automation of Electronic Systems*, vol. 10, pp. 303–329, Apr. 2005.

[28] S. Ozev and C. Olgaard, “Wafer-level RF test and DfT for VCO modulating transceiver architectures,” in *Proceedings of IEEE VLSI Test Symposium*, 2004, pp. 217–222.

[29] A. B. Kahng, “The road ahead: The significance of packaging,” *IEEE Design and Test*, pp. 104–105, Nov. 2002.

[30] W. Lau, “Measurement challenges for on-wafer RF-SOC test,” in *Proceedings of Electronics Manufacturing Technology Symposium*, 2002, pp. 353–359.

[31] R. Brederlow, W. Weber, J. Sauerer, S. Donnay, P. Wambacq, and M. Vertregt, “A mixed-signal design roadmap,” *IEEE Design and Test*, vol. 18, pp. 34–46, Nov. 2001.

[32] G. Bao, “Challenges in low cost test approach for ARM core based mixed-signal SoC DragonBall^{tm}-MX1,” in *Proceedings of International Test Conference*, 2003, pp. 512–519.

[33] J. Sweeney and A. Tsefrekas, “Reducing test cost through the use of digital testers for analog tests,” in *Proceedings of International Test Conference*, 2005, pp. 1–9.

[34] M. Allison, “Wafer probe acquires a new importance in testing,” *IEEE Design & Test of Computers*, vol. 5, pp. 45–49, May. 2005.

[35] A. Singh, P. Nigh, and C. M. Krishna, “Screening for known good die (KGD) based on defect clustering: an experimental study,” in *Proceedings of International Test Conference*, 1997, pp. 362–371.

[36] “Full wafer contact burn-in and test system”, http://www.aehr.com/products/fox_14_data_sheets.pdf.

[37] “Innovative burn-in testing for SoC devices with high power dissipation”, http://www.advantest.de/dasat/index.php?cid=100363&conid=101096&sid=17d2c133fab7783a035471392fd60862.

[38] P. Tadayon, “Thermal challenges during microprocessor testing,” *Intel Technology Journal*, vol. 3, pp. 1–8, 2000.

[39] P. Nigh, "Scan-based testing: The only practical solution for testing asic/consumer products," in *Proceedings of International Test Conference*, 2002.

[40] A. Vassighi, O. Semenov, and M. Sachdev, "Thermal runaway avoidance during burn-in," in *Proceedings of International Reliability Physics Symposium*, 2004, pp. 655–656.

[41] K. Kanda, K. Nose, H. Kawaguchi, and T. Sakurai, "Design impact of positive temperature dependence on drain current in Sub-1V CMOS VLSIs," *IEEE Journal of Solid-State Circuits*, vol. 36, pp. 1559–1564, Oct. 2001.

[42] E. Larsson, J. Pouget and Z. Peng, "Defect-aware SoC test scheduling," in *Proceedings of VLSI Test Symposium*, 2004, pp. 228–233.

[43] U. Ingelsson, S. K. Goel, E. Larsson and E. J. Marinissen, "Test scheduling for modular SOCs in an abort-on-fail environment," in *Proceedings of European Test Symposium*, 2005, pp. 8–13.

[44] R. W. Bassett, B. J. Butkus, S. L. Dingle, M. R. Faucher, P. S. Gillis, J. H. Panner, J. G. Petrovick, and D. L. Wheater, "Low-cost testing of high-density logic components," in *Proceedings of International Test Conference*, 1989, pp. 550–758.

[45] J. Darringer, E. Davidson, D. J. Hathaway, B. Koenemann, M. Lavin, J. K. Morrell, K. Rahmat, W. Roesner, E. Schanzenbach, G. Tellez, and L. Trevillyan, "EDA in IBM: Past, present, and future," *IEEE Transactions on Computer-Aided Design*, vol. 19, pp. 1476–1497, Dec. 2000.

[46] H. F. H. Vranken, T. Waayers and D. Lelouvier, "Enhanced reduced pin-count test for full scan design," in *Proceedings of International Test Conference*, 2001, pp. 738–747.

[47] J. Jahangiri, N. Mukherjee, W. T. Cheng, S. Mahadevan and R. Press, "Achieving high test quality with reduced pin count testing," in *Proceedings of Asian Test Symposium*, 2005, pp. 312–317.

[48] T. G. Foote, D. E. Hoffman, W. V. Huott, T. J. Koprowski, M. P. Kusko, and B. J. Robbins, "Testing the 500-MHz IBM S/390 Microprocessor," *IEEE Design & Test of Computers*, vol. 15, no. 3, pp. 83–89, 1998.

[49] B. Koupal and T. Lee and B. Gravens. Bluetooth Single Chip Radios: Holy Grail or White Elephant, http://www.signiatech.com/pdf/paper two chip.pdf.

[50] C. Pan and K. Cheng, "Pseudo-random testing and signature analysis for mixed-signal circuits," in *Proceedings of International Conference on Computer Aided Design*, 1995, pp. 102–107.

[51] N. A. M. Hafed and G. W. Roberts, "A stand-alone integrated test core for time and frequency domain measurements," in *Proceedings of International Test Conference*, 2001, pp. 1190–1199.

[52] A. Sehgal, F. Liu, S. Ozev, and K. Chakrabarty, "Test planning for mixed-signal SOCs with wrapped analog cores," in *Proceedings of Design Automation and Test in Europe Conference*, 2005, pp. 50–55.

[53] C. Taillefer and G. Roberts, "Reducing measurement uncertainty in a DSP-based mixed-signal test environment without increasing test time," *IEEE Transactions on VLSI Systems*, vol. 13, pp. 862–861, Jul. 2005.

[54] S. Bahukudumbi and K. Bharath, "A low overhead high speed histogram based test methodology for analog circuits and IP cores," in *Proceedings of International Conference on VLSI Design*, 2005, pp. 804–807.

[55] A. Sehgal, S. Ozev, and K. Chakrabarty, "Test infrastructure design for mixed-signal SOCs with wrapped analog cores," *IEEE Transactions on VLSI Systems*, vol. 14, pp. 292–304, Mar. 2006.

[56] M. d'Abreu, "Noise – its sources, and impact on design and test of mixed signal circuits," in *Proceedings of International Workshop on Electronic Design, Test and Applications*, 1997, pp. 370–374.

[57] W. R. Daasch, K. Cota, J. McNames, and R. Madge, "Neighbor selection for variance reduction in IDDQ and other parametric data," in *Proceedings of International Test Conference*, 2001, pp. 1240–1249.

[58] S. Sabade and D. M. H. Walker, "Improved wafer-level spatial analysis for IDDQ limit setting," in *Proceedings of International Test Conference*, 2001, pp. 82–91.

[59] A. Keshavarzi, K. Roy, C. F. Hawkins, and V. De, "Multiple-parameter CMOS IC testing with increased sensitivity for IDDQ," *IEEE Transactions on VLSI Systems*, vol. 11, pp. 863–870, Oct. 2003.

[60] Y. Zorian, "A distributed BIST control scheme for complex VLSI devices," in *Proceedings of VLSI Test Symposium*, 1993, pp. 4–9.

[61] S. Wang and S. K. Gupta, "An automatic test pattern generator for minimizing switching activity during scan testing activity," *IEEE Transactions on Computer-Aided Design*, vol. 21, pp. 954–968, Aug. 2002.

[62] P. Girard, "Survey of low-power testing of VLSI circuits," *IEEE Design & Test of Computers*, vol. 19, pp. 80–90, May 2002.

[63] R. Sankaralingam, R. R. Oruganti, and N. A. Touba, “Static compaction techniques to control scan vector power dissipation,” in *Proceedings of VLSI Test Symposium*, 2000, pp. 35–40.

[64] K. M. Butler, J. Saxena, A. Jain, T. Fryars, J. Lewis, and G. Hetherington, “Minimizing power consumption in scan testing: pattern generation and DFT techniques,” in *Proceedings of International Test Conference*, 2004, pp. 355–364.

[65] J. Saxena, K. M. Butler, and L. Whetsel, “An analysis of power reduction techniques in scan testing,” in *Proceedings of International Test Conference*, 2001, pp. 670–677.

[66] X. Wen, Y. Yamashita, S. Kajihara, L. T. Wang, K. K. Saluja, and K. Kinoshita, “On low-capture-power test generation for scan testing,” in *Proceedings of VLSI Test Symposium*, 2005, pp. 265–270.

[67] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. M. Reddy, “Techniques for minimizing power dissipation in scan and combinational circuits during test application,” *IEEE Transactions on Computer-Aided Design*, vol. 17, pp. 1325–1333, Dec. 1998.

[68] P. K. Latypov, “Energy saving testing of circuits,” *Automation and Remote Control*, vol. 62, pp. 653–655, Apr. 2001.

[69] “Synopsys TetraMAX ATPG methodology backgrounder”, www.synopsys.com/products/test/tetramax_wp.html/.

[70] W. Li, S. M. Reddy, and I. Pomeranz, “On reducing peak current and power during test,” in *Proceedings of ISVLSI*, 2005, pp. 156–161.

[71] X. Wen, Y. Yamashita, S. Morishima, S. Kajihara, L. T. Wang, K. K. Saluja, and K. Kinoshita, “Low-capture-power test generation for scan-based at speed testing,” in *Proceedings of International Test Conference*, 2005, pp. 1019–1028.

[72] S. Bahukudumbi and K. Chakrabarty, “Defect-oriented and time-constrained wafer-level test length selection for core-based digital SoCs,” in *Proceedings of International Test Conference 2006*, Oct. 2006, pp. 1–10.

[73] ——, “Wafer-level modular testing of core-based SOCs,” *IEEE Transactions on VLSI Systems*, vol. 15, pp. 1144–1154, Oct. 2007.

[74] ——, “Test-length selection, reduced pin-count testing, and tam optimization for wafer-level testing of core-based digital SoCs,” in *Proceedings of International Conference on VLSI Design*, 2007, pp. 459–464.

[75] I. Koren, Z. Koren, and C. H. Strapper, “A unified negative-binomial distribution for yield analysis of defect-tolerant circuits,” *IEEE Transactions on Computers*, vol. 42, pp. 724–734, Jun. 1993.

[76] I. Koren and C. H. Strapper, *Yield Models for Defect Tolerant VLSI circuit: A Review.* Plennum, 1989.

[77] C. H. Strapper, “Small-area fault clusters and fault-tolerance in VLSI systems,” *IBM Journal on Research and Development*, vol. 33, pp. 174–177, Mar. 1989.

[78] J. A. Cunningham, “The use and evaluation of yield models in integrated circuit manufacturing,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 3, pp. 60–71, May 1990.

[79] T. S. Barnett, M. Grady, K. G. Purdy, and A. D. Singh, “Combining negative binomial and weibull distributions for yield and reliability predictions,” *IEEE Design & Test of Computers*, vol. 23, pp. 110–116, December 2006.

[80] T. S. Barnett and A. D. Singh, “Relating yield models to burn-in fall-out in time,” in *Proceedings of International Test Conference*, 2003, pp. 77–84.

[81] J. T. de Sousa and V. D. Agrawal, “Reducing the complexity of defect level modeling using the clustering effect,” in *Proceedings of Design Automation and Test in Europe Conference*, 2000, pp. 640–644.

[82] S. K. Goel and E. J. Marinissen, “Layout-driven SoC test architecture design for test time and wire length minimization,” in *Proceedings of Design Automation and Test in Europe Conference*, 2003, pp. 10 738–10 743.

[83] E. J. Marinissen, V. Iyengar and K. Chakrabarty, “A set of benchmarks for modular testing of SOCs,” in *Proceedings of International Test Conference*, 2002, pp. 519–528.

[84] E. Larsson and Z. Peng, “An integrated framework for the design and optimization of SoC test solutions,” *Journal of Electronic Testing: Theory and Applications*, vol. 18, pp. 385–400, Feb. 2002.

[85] V. Iyengar and K. Chakrabarty, “Test bus sizing for system-on-a-chip,” *IEEE Transactions on Computers*, vol. 51, pp. 449–459, May 2005.

[86] E. Larsson and H. Fujiwara, “Optimal system-on-chip test scheduling,” in *Proceedings of Asian Test Symposium*, 2004, pp. 306–311.

[87] E. Kreyszig, *Advanced Engineering Mathematics*, 8th ed. John Wiley & Sons Inc., 1998.

[88] M. Berkelaar et al., “lpsolve: Open source (mixed-integer) linear programming system”. Version 5.5 dated May 16, 2005
URL: http://www.geocities.com/lpsolve.

[89] Frontline Systems Inc., Incline Village, NV,“Premium Solver Platform” 2007. [Online]. Available: http://www.solver.com/xlsplatform.html.

[90] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms*, 3rd ed. Wiley, 2006.

[91] R. Webster, *Convexity*, 2nd ed. Oxford Science Publications, 1995.

[92] K. Chakrabarty, V. Iyengar, and M. D. Krasniewski, “Test planning for modular testing of hierarchical SOCs,” *IEEE Transactions on Computer-Aided Design*, vol. 24, pp. 435–448, Mar. 2005.

[93] S. Boyd, S. J. Kim, L. Vandenberghe, and A. Hassibi, “A tutorial on geometric programming,” *Optimization and Engineering*, vol. 8, pp. 67–127, Apr. 2007.

[94] “TOMLAB Optimization: TOMLAB/GP”
URL: http://www.tomopt.com.

[95] A. Cron, “IEEE P1149.4– almost a standard,” in *Proceedings of International Test Conference*, 1997, pp. 174–182.

[96] S. Bernard, M. Comte, F. Azais, Y. Bertrand, and M. Renovell, “A new methodology for ADC test flow optimization,” in *Proceedings of International Test Conference*, 2003, pp. 201–209.

[97] S. Bahukudumbi, S. Ozev, K. Chakrabarty, and V. Iyengar, “A wafer-level defect screening technique to reduce test and packaging costs for ”big-D/small-A” mixed-signal SoCs,” in *Proceedings of Asia South Pacific Design Automation*, 2007, pp. 823–828.

[98] A. Frisch and T. Almy, “HABIST: histogram based analog built in self test,” in *Proceedings of International Test Conference*, 1997, pp. 760–767.

[99] E. Acar and S. Ozev, “Delayed-RF based test development for fm transceivers using signature analysis,” in *Proceedings of International Test Conference*, 2004, pp. 783–792.

[100] S. K. Sunter and N. Nagi, “A simplified polynomial-fitting algorithm for DAC and ADC BIST,” in *Proceedings of International Test Conference*, 1997, pp. 389–395.

[101] T. Kuyel, “Linearity testing issues of analog to digital converters,” in *Proceedings of International Test Conference*, 1999, pp. 747–756.

[102] U. Ingelsson, S. K. Goel, E. Larsson, and E. J. Marinissen, "Test scheduling for modular SOCs in an abort-on-fail environment," in *Proceedings of European Test Symposium*, 2005, pp. 8–13.

[103] D. E. Becker and A. Sandborn, "On the use of yielded cost in modeling electronic assembly processes," *IEEE Transactions on Electronics Packaging Manufacturing*, vol. 24, pp. 195–202, Jul. 2001.

[104] S. Edbom and E. Larsson, "An integrated technique for test vector selection and test scheduling under test time constraint," in *Proceedings of Thirteenth Asian Test Symposium*, 2004, pp. 254–257.

[105] G. Chen, S. M. Reddy and I. Pomeranz, "Procedures for identifying untestable and redundant transition faults in synchronous sequential circuits," in *Proceedings of International Conference on Computer Design*, 2003, pp. 36–41.

[106] http://www.mosis.org.

[107] M. Shen, Z. Li-Rong, and H. Tenhunen, "Cost and performance analysis for mixed-signal system implementation: System-on-chip or system-on-package," *IEEE Transactions on Electronics Packaging Manufacturing*, vol. 25, pp. 522–545, Oct. 2002.

[108] S. Bahukudumbi, K. Chakrabarty, and R. Kacprowicz, "Test scheduling for wafer-level test-during-burn-in of core-based SoCs," in *Proceedings of Design Automation and Test in Europe Conference*, 2008, to appear.

[109] S. Samii, E. Larsson, K. Chakrabarty, and Z. Peng, "Cycle-accurate test power modeling and its application to SOC test scheduling," in *Proceedings of International Test Conference*, 2006.

[110] D. B. West, *Introduction to Graph Theory.* Prentice Hall, 2000.

[111] V. Iyengar, K. Chakrabarty, and E. J. Marinissen, "Test wrapper and test access mechanism co-optimization for system-on-chip," *Journal of Electronic Testing: Theory and Applications*, vol. 18, pp. 213–230, Apr. 2002.

[112] M. Garey and D. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[113] IWLS 2005 Benchmarks, "http://iwls.org/iwls2005/benchmarks.html."

[114] M. E. Imhof, C. G. Zoellin, H. Wunderlich, N. Maeding, and J. Leenstra, "Scan test planning for power reduction," in *Proceedings of Design Automation Conference*, 2007, pp. 521–526.

[115] P. M. Rosinger, B. M. Al-Hashimi, and N. Nicolici, “Power profile manipulation: A new approach for reducing test application time under power constraints,” *IEEE Transactions on Computer-Aided Design*, vol. 21, pp. 1217–1225, May 2002.

[116] J. Costa, P. F. Flores, H. C. Neto, J. C. Monteiro, and J. P. Marques-Silva, “Exploiting don’t cares in test patterns to reduce power during BIST,” in *Proceedings of European Test Workshop*, 1998, pp. 34–36.

[117] S. Ghosh, S. Basu, and N. A. Touba, “Joint minimization of power and area in scan testing by scan cell reordering,” in *Proceedings of Annual Symposium on VLSI*, 2003, pp. 246–249.

[118] Z. Zhang, S. M. Reddy, I. Pomeranz, J. Rajski, and B. M. Al-Hashimi, “Enhancing delay fault coverage through low power segmented scan,” in *Proceedings of European Test Symposium*, 2006, pp. 21–28.

[119] G. L. Vairaktarakis, “On Gilmore-Gomory’s open question for the bottleneck tsp,” *Computers & Operations Research*, vol. 31, pp. 483–491, Nov. 2003.

[120] R. Y. Rubinstein and D. P. Kroese, *A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning.* Springer-Verlag New York, LLC, 2004.

[121] A. Benso, A. Bosio, S. D. Carlo, G. D. Natale, and P. Prinetto, “ATPG for dynamic burn-in test in full-scan circuits,” in *Proceedings of Asian Test Symposium*, 2006, pp. 75–82.

[122] T. Cooper, G. Flynn, G. Ganesan, R. Nolan, and C. Tran, “Demonstration and deployment of a test cost reduction strategy using design-for test (DFT) and wafer level burn-in and test,” *Future Fab*, vol. 11, Jun. 2001.

[123] S. Bahukudumbi and K. Chakrabarty, “Test-pattern ordering for wafer-level test-during-burn-in,” in *Proceedings of VLSI Test Symposium*, 2008, to appear.

[124] R. Sankaralingam and N. A. Touba, “Controlling peak power during scan testing,” in *Proceedings of VLSI Test Symposium*, 2002, pp. 153–159.

[125] J. E. Nelson, T. Zanon, R. Desineni, J. G. Brown, N. Patil, W. Maly, and R. D. Blanton, “Extraction of defect density and size distributions from wafer sort test results,” in *Proceedings of Design Automation and Test in Europe Conference*, 2006, pp. 913–918.

[126] Z. Wang and K. Chakrabarty, “Test-quality/cost optimization using output-deviation-based reordering of test patterns,” *IEEE Transactions on Computer-Aided Design*, vol. 27, pp. 352–365, Feb. 2008.

[127] D. Ghosh, S. Bhunia, and K. Roy, “Multiple scan chain design technique for power reduction during test application in BIST,” in *Proceedings of International Symposium on Defect and Fault Tolerance in VLSI Systems*, 2003, pp. 191–198.

[128] N. Nicolici and B. M. Al-Hashimi, “Multiple scan chains for power minimization during test application in sequential circuits,” *IEEE Transactions on Computers*, vol. 51, pp. 721–733, Jun. 2002.

[129] “FLOTHERM: Design-Class Thermal Analysis for Electronics”
URL: http://www.flomerics.com/products/flotherm/.

[130] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, “Efficient scan chain design for power minimization during scan testing under routing constraint,” in *Proceedings of International Test Conference*, 2003, pp. 488–493.

[131] K. Sankaranarayanan, S. Velusamy, M. Stan, and K. Skadron, “A case for thermal-aware floorplanning at the microarchitectural level,” *Journal of Instruction-Level Parallelism*, vol. 8, pp. 8–16, Oct 2005.

Biography

Sudarshan Bahukudumbi spb@ee.duke.edu

PERSONAL DATA

Date of birth: April 4, 1982.
Place of birth: Chennai, Tamil Nadu, India.

EDUCATION

Doctor of Philosophy, Duke University, USA, expected 2008.
Master of Science, New Mexico State University, USA, 2005.
Bachelor of Engineering, University of Madras, India, 2003.

PUBLICATIONS

- **Journal Articles**

  1. Sudarshan Bahukudumbi and Krishnendu Chakrabarty, "Wafer-level modular testing of core-based SOCs", *IEEE Transactions on VLSI Systems*, vol. 15, October 2007, pp. 1144–1154.
  *2. A. Sehgal, S. Bahukudumbi and K. Chakrabarty, "Power-aware SOC test planning for effective utilization of port-scalable testers", accepted for publication in *ACM Transactions on Design Automation of Electronic Systems*.
  3. S. Bahukudumbi, S. Ozev, K. Chakrabarty and V. Iyengar, "Wafer-level defect screening for "big-D/small-A" mixed-signal SoCs", accepted for publication in *IEEE Transactions on VLSI Systems*.

- **Refereed Conference Papers**

  1. S. Bahukudumbi and Krishnendu Chakrabarty, "Defect-oriented and time-constrained wafer-level test length selection for core-based SOCs", *Proc. IEEE International Test Conference*, 2006.
  2. S. Bahukudumbi and Krishnendu Chakrabarty, "Test-length selection, reduced pin-count testing, and TAM optimization for wafer-level testing of core-based digital SoCs", *Proc. IEEE International Conference on VLSI Design*, pp. 459-464, 2007.
  3. S. Bahukudumbi, S. Ozev, K. Chakrabarty and V. Iyengar, "A wafer-level defect screening technique to reduce test and packaging costs for "big-D/small-A" mixed-signal SoCs", *Proc. IEEE/ACM Asia South Pacific Design Automation Conference*, pp. 823-828, 2007.

4. S. Bahukudumbi, K. Chakrabarty and R. Kacprowicz, "Test scheduling for wafer-level test-during-burn-in of core-based SoCs", *Proc. Design Automation and Test in Europe (DATE) Conference*, pp. 1103-1106, 2008.
5. S. Bahukudumbi and K. Chakrabarty, "Test-pattern ordering for wafer-level test-during-burn-in", accepted for publication in *Proc. IEEE VLSI Test Symposium*, 2008.
6. [*] S. Bahukudumbi and K. Bharath, "A low overhead high speed histogram based test methodology for analog circuits and IP Cores", *Proc. IEEE International Conference on VLSI Design*, pp. 804-807, 2005.
7. [*] P. Srivatsan, S. Bahukudumbi and P. P. Bhaskaran, "DYNORA: A new caching technique", *Proc. IEEE Euromicro Symposium on Digital Systems Design*, pp. 70-75, 2003.

- **Submitted Papers**

  1. S. Bahukudumbi and K. Chakrabarty, "Power management for wafer-level test during burn-in", submitted to *IEEE International Test Conference*, 2008.

Professional Activities

- IEEE student member
- Served as a reviewer for the IEEE Transactions on VLSI, IEEE Transactions on CAD and the International Test Conference.

---

[*] Not related to Ph.D. thesis work.