

Technical Report on

Embedded Testing Architectures:  
Literature, Open Issues and Methods

Vasileios Tenentes, Xrysovalantis Kavousianos  
Department of Computer Science & Engineering  
University of Ioannina

Technical Report Number: 02 - 2013

*This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) – Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.*



**European Union**  
European Social Fund



OPERATIONAL PROGRAMME  
EDUCATION AND LIFELONG LEARNING  
MINISTRY OF EDUCATION & RELIGIOUS AFFAIRS  
MANAGING AUTHORITY



**NSRF**  
2007-2013  
programme for development  
EUROPEAN SOCIAL FUND

Co- financed by Greece and the European Union

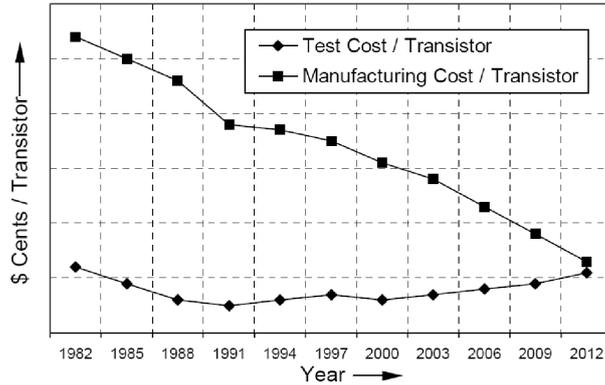


Figure 1: ITRS'07 test cost predictions

## 1 Introduction

Nowadays, Very-Deep-Sub-Micron (VDSM) integration technology is in our everyday life with portable Multi-Core Systems-on-Chips (MCSoCs) that contain billions of transistors. However, even from their first construction, in the 1960s, the integrated circuits (ICs), commonly referred to as microchips or simply chips, were accompanied by the need of testing. Tens of transistors integrated into Small-scale integration (SSI) devices in the early 1960s and hundreds of transistors integrated into medium-scale integration (MSI) devices in the late 1960s, were relatively simple to test. However, thousands and tens of thousands of transistors integrated into large-scale integration (LSI) devices in the 1970s and hundreds of thousands of transistors integrated into very-large-scale integration (VLSI) devices in the early 1980s introduced serious test challenges.

This trend of higher integration scaling over the years, generally known as Moore's law [77], is the result of the exponential decrease in a transistor's manufacturing cost. However, the test cost of a transistor does not share the same trend. Figure 1 illustrates the test versus manufacturing cost of a transistor over the years as reported in 2007's International Technology Roadmap of Semiconductors (ITRS) [2]. These news were alarming and IC testing emerged as a potential bottleneck for future exponential integration scaling according to Moore's law.

This report presents the motivation for IC testing and the major drivers that affect test cost. It introduces scan design, which is the most widely used manufacturing testing architecture, and presents the new architectural and designing trends that affect test cost.

## 2 Structural Manufacturing Testing

There are many techniques developed over the years for IC's manufacturing testing, but the most widely adopted one, that offers the lowest reject rate versus test cost, is *structural testing*. In this report, the basic concepts of structural testing are introduced.

A *fault* is a representation of a defect reflecting a physical condition that causes a circuit to fail to perform as designed. A *failure* is a deviation in the performance of

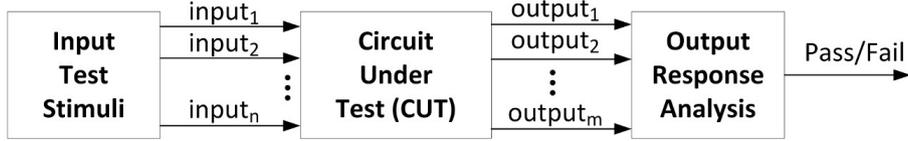


Figure 2: Basic Testing Approach.

a circuit or system from its specified behavior and represents an irreversible state of a component such that it must be repaired in order for it to provide its intended design function. A *circuit error* is a wrong output signal produced by a defective circuit. A circuit defect may lead to a fault, a fault can cause a circuit error, and a circuit error can result in a system failure [121].

During testing a set of test stimuli (referred also as test vectors or test patterns) is applied to the  $n$  inputs of the CUT, and its  $m$  output responses are analyzed, as illustrated in Figure 2. Circuits that produce the correct output responses for all input stimuli pass the test and are considered to be defect-free. Those circuits that fail to produce a correct response at any point during the test sequence are assumed to be defective.

The ultimate target of any ICs test mechanism is to test the chips for all possible defects, or in other words, to achieve complete defect coverage. However, such a goal is not realistic, and thus fault models are adopted. Fault models save time and improve test efficiency, as a limited number of test patterns that target specific faults, related to the structure of the CUT, are applied at the circuit’s inputs. This process is called structural testing. Any input pattern (test stimuli), that produces a different output response in a faulty circuit from that of the fault-free circuit is a *test vector* that will detect the faults. Any set of *test vectors* is called a *test set*. The goal of Automatic Test Patterns Generation (ATPG) tools is to find an efficient test set that detect as many defects as possible for a given CUT and a given fault model. These tools provide a quantitative measure of the fault-detection capabilities of a given test set for a targeted fault model. This measure is called fault coverage and is defined as:

$$\text{Fault coverage} = \frac{\text{Number of detected faults}}{\text{Total number of faults}}$$

Fault coverage is linked to the *quality* of a manufacturing process, which is expressed by the *yield*, and the *quality* of a the testing process, which is expressed by the *reject rate*, by the following relation [130]:

$$\text{Reject rate} = 1 - \text{yield}^{(1-\text{fault coverage})}$$

From this equation, we can show that a SoC with 40 cores, each having 90% fault coverage and 90% yield, could result in a reject rate of 41.9%, or 419,000 PPM. As a result, improving fault coverage can be easier and less expensive than improving manufacturing yield because making yield enhancements can be costly. Therefore, generating test stimuli with high fault coverage is very important.

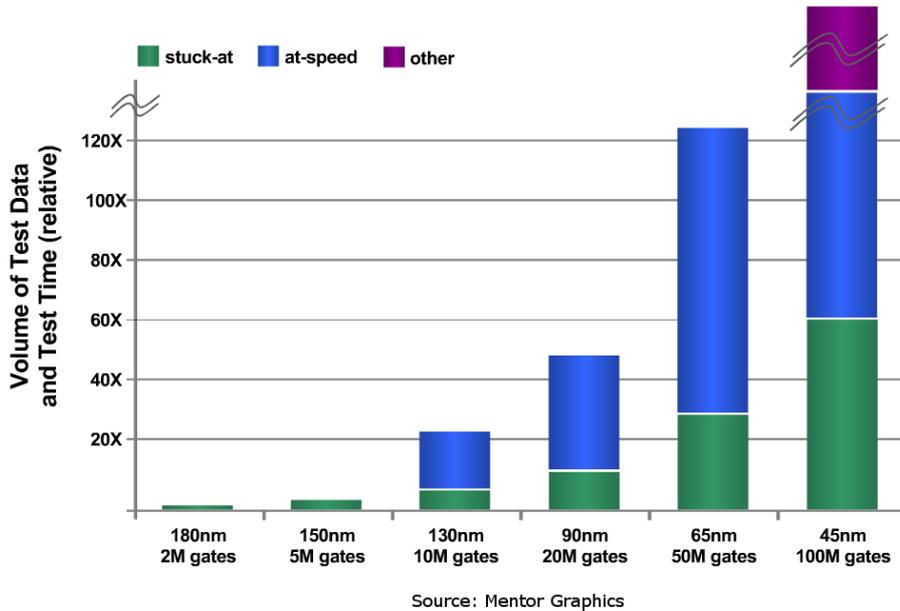


Figure 3: Explosion of test data volume.

Unfortunately, structural testing has its own limitations too. Fault models are used as an abstraction description of possible defects on a given design structure.

- A single fault model cannot cover all possible defects. To overcome this limitation, industry uses multiple fault models.
- Even when defects can be modeled by a fault model, sometimes it is impossible to get 100% fault coverage due to testability limitations caused by either the structure of the CUT or by the way the test is conducted (undetected faults<sup>1</sup>).

### 3 Basic Test Cost Factors

Beside the quality enhancement of structural testing methods that indirectly reduce test cost, new testing techniques should also consider the classical test cost factors. Since, the market demands require faster and denser ICs over the years, these basic cost factors have been stressed by the new dense and complex integration technologies. These factors are the cost and the limitations of ATEs, the time required to perform testing and unpredictable human factors.

**Equipment Cost:** The major contributor to the cost of testing is the cost of the ATEs. As devices continue to grow more complex, the test capabilities need to be constantly improved. Also, the speed of the ATE is required to increase because constant device scaling since the mid-1980s has pushed the device speeds significantly higher. Manufacturers are constantly looking for low-cost ATEs that can reliably test complex and high-speed device

<sup>1</sup>An undetectable fault occurs where there is no test to distinguish the fault-free circuit from a faulty circuit containing that fault.

during high-volume production testing.

**ATE Limitations:** The ever-increasing number of gates results in an ever-increasing number of test patterns. The 2007 ITRS test report [2] predicted that the test-data volume for integrated circuits will be as much as 38 times higher and the test-application time will be about 17 times larger in 2015 than it was in 2007. Figure 3 captures this trend. While test data increase, the previous ATEs generations cannot cope with the demanding memory and CUT/ATE communication requirements. All these result to the following ATE limitations:

- Bandwidth limitations between Workstation/ATE: the test patterns need to be uploaded from the workstation to the ATE memory. Limited data bandwidth between workstation/ATE may stall this process from several tens of minutes to hours [40]. While ATE remains idle, the cost of test increases [116].
- ATE memory limitations: New generation ATEs with the required memory may not be available (or may be very expensive). Test data are truncated to fit the memory, resulting to quality degradation [40].
- Bandwidth limitations between ATE/CUT: To apply the test patterns at the CUT, they need to be transferred from the ATE (where they are stored) to the CUT. Additionally, the responses must pass from the CUT to the ATE in order to be analyzed. The bandwidth of the channels between ATE/CUT are limited. The above process may increase dramatically test time and consequently test cost [116].

**Production Test Time:** Apart from the cost of ATEs, large test application time is a major factor for increased test costs. Typically, test time for wireless devices ranges from a few seconds to a few minutes. During production, when millions of devices are tested, even such apparently small test times can create a bottleneck. Suppose, for example, that a device test time required during production is 60 seconds. Therefore, the number of devices that can be tested is 1440 per day ( $= 24 \times 3600/60$ ). Considering that 10 ATEs are used, then to release a million devices to the market requires 70 days. This clearly shows that a small reduction in test time can increase the throughput significantly. Therefore, there is a constant need in the test community to reduce production test time. Production test time is affected by many factors, such as the time needed to design the tests (by the test engineer), the time required for the equipment (handlers and probers) to prepare the environment for the test, the Test Application Time (TAT), which is the time needed to excite the CUT with the test stimuli and get the responses.

**Human factor:** Additional costs come from engineering errors or other human factors. For example, an improperly designed IC or a bug in the test program can significantly increase the time required to release a product. This can cause the manufacturer to lose significant market share for that product. Such factors can be fatal for small businesses, and the success of the manufacturer relies heavily on the test process.

In general, all the above limitations (except human factor) stem from the same reason: the **increasing amount of test data** (stimulus and response data) [114, 116]. The

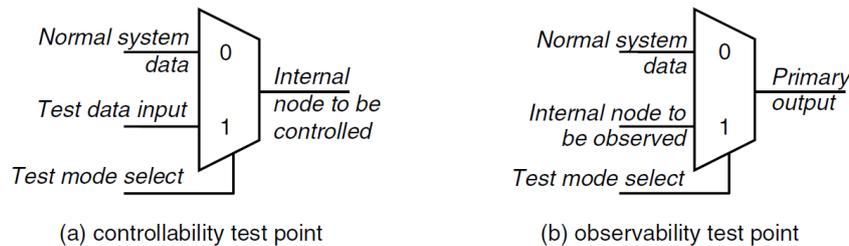


Figure 4: DFT test point

test cost solution to this problem is to often upgrade ATEs, but this solution is very impractical and extremely costly to be adopted by companies. The necessity to overpass this dead end, decrease test cost and handle the increased complexity of new integration technologies, motivated the consideration of testing during the early life of manufacturing ICs: the design. It was the dawn of Design for Testability (DFT).

## 4 Design for Testability

Test engineers usually have to construct test vectors after the design is completed. This invariably requires a substantial amount of time and effort that could be avoided if testing was considered early in the design flow to make the design more testable. As a result, integration of design and test, referred to as design for testability (DFT), was proposed in the 1970s.

To test the structure of ICs, we need to control and observe logic values of internal nodes. Unfortunately, some nodes in sequential circuits can be very difficult to control and observe; for example, activity on the most significant bit of an  $n$ -bit counter can only be observed after  $2^{n-1}$  clock cycles. Testability measures of controllability and/or observability were first defined in the 1970s [34] to help find those parts of a digital circuit that will be most difficult to test and to assist in test pattern generation for fault detection. Many DFT techniques have been proposed since that time [74]. DFT techniques generally fall into one of the following three categories: (1) ad-hoc DFT techniques, (2) scan design, or (3) built-in self-test (BIST).

### 4.1 Ad-hoc DFT

Ad-hoc methods were the first DFT techniques introduced in the 1970s. The goal was to target only those portions of the circuit that would be difficult to test and to add circuitry to improve the controllability or observability. Ad-hoc techniques typically use test point insertion to access internal nodes directly. An example of a test point is a multiplexer inserted to control or observe an internal node, as illustrated in Figure 4.

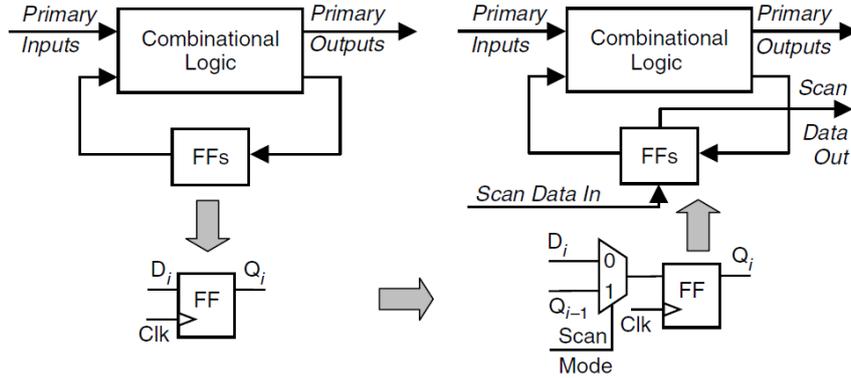


Figure 5: Adding test points at a sequential circuit.

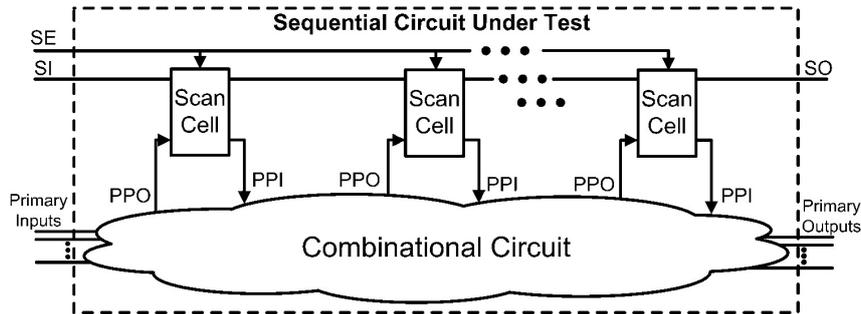


Figure 6: A typical scan design.

## 4.2 Scan Design

In scan design [28] external access is provided at the storage elements of ICs in order to increase their controllability and observability. The modified storage elements are commonly referred to as *scan cells*. Once the capability of controlling and observing the internal states of a design is added, the problem of testing a sequential circuit is transformed into a problem of testing combinational logic, which is an easier task. Figure 5 presents the re-designing of D flip-flops for a sequential circuit to scan cells. Widely used scan cell designs are: muxed-D scan cell, clocked-scan cell [74], and level-sensitive scan design (LSSD) cell [26, 28].

In order to save I/O pins, the scan cells are connected into multiple shift registers, called *scan chains*. A typical scan design, with a single scan chain, is presented in Figure 6. Scan design accomplishes this task by replacing all selected storage elements with scan cells, each having one additional scan input (SI) port and one shared/additional scan output (SO) port. By connecting the SO port of one scan cell to the SI port of the next scan cell a scan chain is created. This way a sequential CUT is transformed into a combinational circuit. The control points of the combinational circuit are called pseudorandom primary inputs (PPIs) and the observable points are called pseudorandom primary outputs (PPOs). The selection between operations of a typical scan design (scan or normal operation modes) is controlled by a scan enable (SE) signal. Testing based on scan design is called *scan testing* and is conducted as follows:

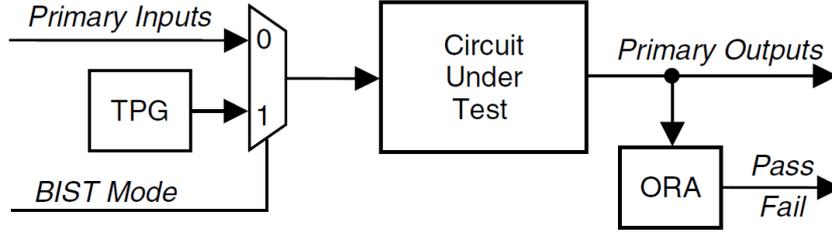


Figure 7: BIST scheme.

- During the scan mode (when SE='1'), the scan chain is used to shift in (or scan in) a test vector to be applied to the combinational logic.
- During one clock cycle in the system mode (when SE='0' and it is also called capture mode) of operation, the test vector is applied to the combinational logic and the output responses are clocked into the flip-flops.
- Also in scan mode, the scan chain is used to shift out (or scan out) the combinational's logic output response to the test vector while shifting in the next test vector to be applied.

### 4.3 Built-In Self-Test

Built-in self-test (BIST) was proposed around the 80s [84, 103, 104]. The basic idea is to integrate a test-pattern generator (TPG) and an output response analyzer (ORA) together with the CUT in order to perform testing internal, as illustrated in Figure 7, without any need of external tester. Since an external tester is not required, BIST reduces considerably test cost. However, there are many challenges in making a design BIST-ready: efficient logic BIST structures must be integrated that should achieve high test quality. However, there are different efficient BIST architectures [32, 33, 113] based on the nature of the logic inside the CUT. A constant problem remains the automation of the BIST-architecture design with the ICs design without impacting the overall product schedule. In [40] it was shown that with automation of the designing process and with constant upgrade of this automation, BIST can become viable for large industrial designs.

### 4.4 Test Resource Partitioning

Test Resource Partitioning (TRP) is a DFT approach for highly dense ICs that decreases test cost by easing the burden of outdated ATE systems. TRP focuses on transferring test functionalities from the ATE towards the CUT. The basic idea is to compress large volumes of test data to small test sets that fit in the memory of an ATE and they are based on a hybrid scan design/built-in-self-test (BIST) approach. The test data are stored on the ATE in a compressed form downloaded at the CUT where they are decompressed and applied. After their application, the responses are compressed on the CUT before they are sent back to the ATE in compressed form.

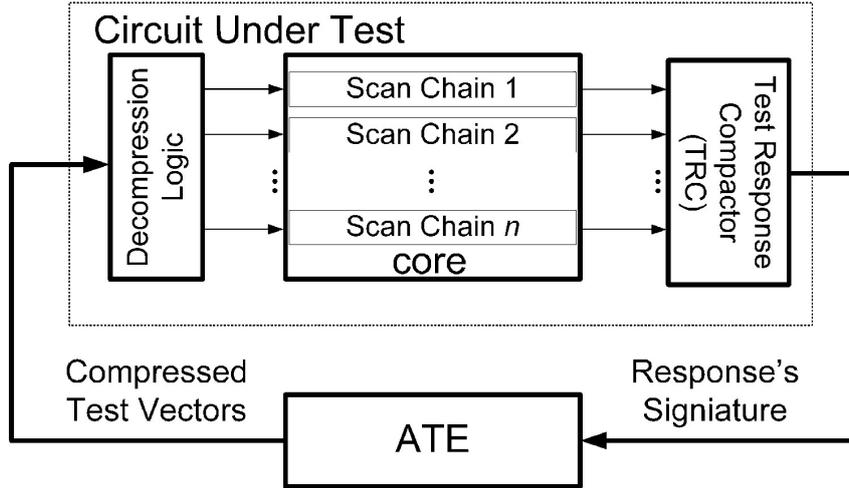


Figure 8: Test Resource Partitioning Architecture

Figure 8 presents the general TRP architecture. The compressed form of test vectors stored into the ATE are called test data. The size of test data, which is the amount of memory required to store the test data on ATE, is called Test Data Volume (TDV). During testing, the test data are transferred through the low-bandwidth ATE/CUT channels to the CUT where they are decompressed on-chip by embedded decompression architectures. The test vectors are shifted into the scan chains setting the CUT into a predetermined internal state. Afterwards, the CUT is let to operate normally and the response is captured into the scan chains. Then, the procedure starts over, but now with the decompression of the next vector. During the shift-in of the next vector, the responses of the previous vector already contained into the scan chains are shift-out towards the TRC where a unique signature for them is created. Signatures are shifted out towards the ATE, where they are compared against fault-free signatures.

The amount of ATE's participation in the testing procedure is the key of categorizing a compression TRP technique. Thus, there are two categories of TRP techniques:

- Test Set Embedding (TSE): long pseudorandom sequences are generated on-chip with minimum interaction with the ATE. TSE techniques have small ATE's memory requirements but they impose large hardware overhead on the embedded decompression architectures and long TAT.
- Test Data Compression (TDC): compression codes, such as statistical, the Run-length, the Golomb, the Frequency-Directed Run-length (FDR) coding, the Huffman code, are utilized to compress the test data. These methods occupy relatively small ATE's memory space, which however is higher than that of TSE techniques, and they also require more frequent usage of the ATE/CUT channels. On the other hand the hardware overhead of the decompressors is very low and the TAT is very short.

TRP techniques are further categorized based on the nature of both the compression code and decompression logic used. There are TRP techniques based on:

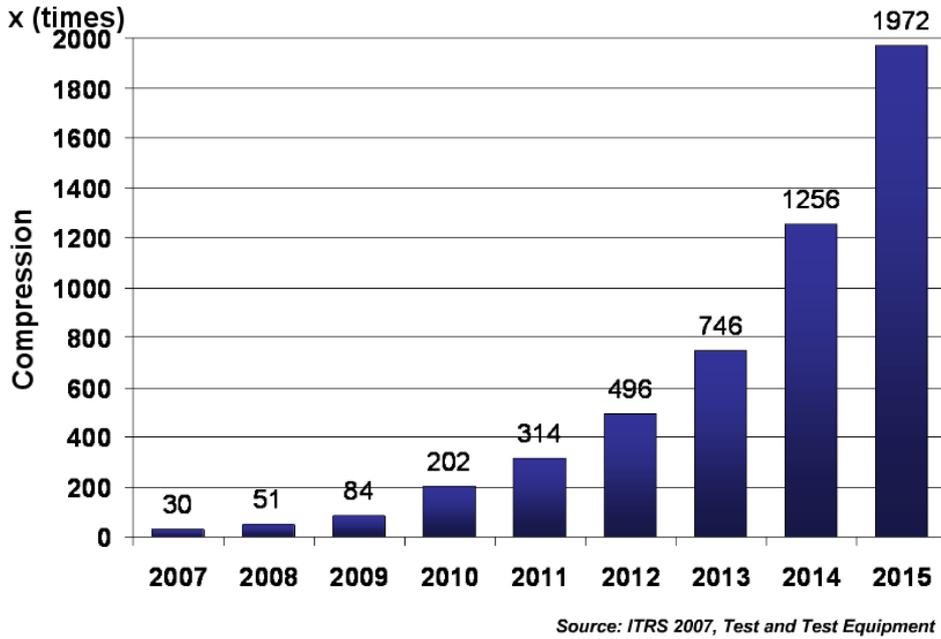


Figure 9: ITRS'07 compression prediction requirements

- Compression codes (code-based): the Golomb [41], the Huffman [42], the Run-length etc [6, 12, 14, 15, 35, 45–47, 53–56, 56, 67, 69, 72, 82, 91, 99, 106, 107, 123, 132–134].
- Linear decompressors (linear-based): Linear Feedback Shift Registers (LFSRs), Ring generators etc [7, 39, 51, 58, 60, 62, 63, 75, 98, 108, 117].
- Broadcast schemes: pseudorandom values broadcasted simultaneously into the scan chains [36, 66, 76, 83, 95, 100, 102, 105, 119, 120].

Commercial tools for test compression are also available [5, 59, 88]. The most widely TRP techniques are based on linear decompressors.

Figure 9 depicts the trajectory in compression requirements of contemporary TRP methods in order to cope with the upcoming explosion of test data. The  $y$ -axis, which is the compression requirements between 2007 and 2015, shows the ratio of uncompressed test data to compressed test data. Since, the existing TRP techniques cannot achieve those compression requirements, new TRP techniques are needed.

The efficiency of TRP's compression is achieved so far, by exploiting only one out of two properties of the test vectors. Specifically, the test vectors consist of logic values '0', '1' as well as *undefined* values 'x'es. The undefined values can be any logic value ('0' or '1') without affecting the stimulation of the fault that the test vector was generated for (undefined values 'x'es are also referred as *don't cares* or *unspecified* values, and logic values '0' and '1' are referred as *defined* or *specified* values in the literature). When a vector contains undefined values it is called *test cube*. The ratio between specified and unspecified values of a test set is called *fill rate*. The two properties that TRP techniques exploit to offer compression are the low fill rate of the test sets (the large amount of unspecified values 'x'es) and the correlation of the specified values that stem from CUT's structural correlation [110]. Linear-based methods exploit the unspecified values, while

code-based techniques exploit the correlations. There is not such technique, so far, that exploits both these properties for compression [110].

## 5 Additional Test Challenges

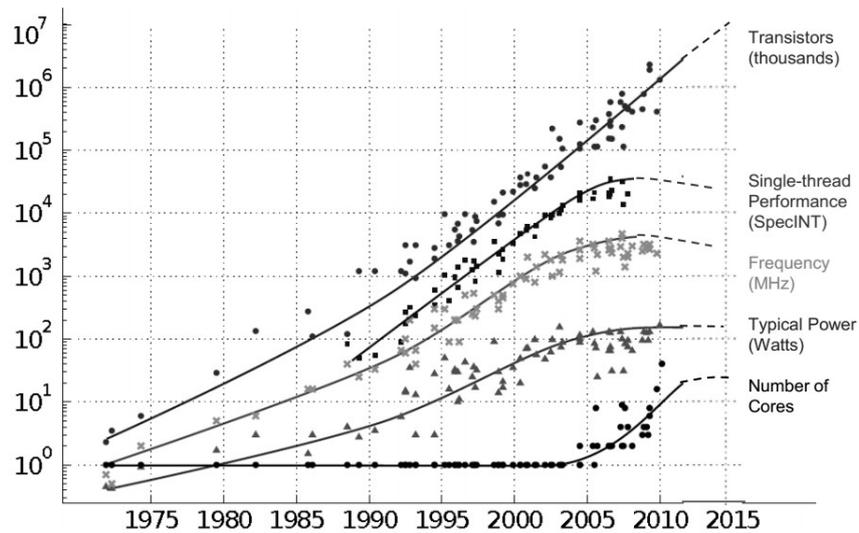
### 5.1 The post-Dennard Era: Low-power Testing

During the last years we have witness a tremendous change in the industry’s target group, since individuals, and not corporations and government agencies, are nowadays the main consumers of semiconductors. During this era the demands for testable low-power mobile devices have been increased dramatically. Nowadays we can find mobile Internet devices (MIDs), personal digital assistants (PDAs) and smartphones which are mobile multimedia-capable devices with wireless Internet access: “supercomputers of older eras in consumers’ “pockets”. The manufacturing of these portable computing devices became reality because of the huge density and speed of contemporary ICs. During these years we also witnessed the transformation of testing ICs to *low-power testing* ICs.

**Density and speed** of ICs have increased exponentially for several decades, following a trend described by Moore’s Law. The original version of Moore’s law states that transistor density doubles every 18-24 months. Although Moore’s law still holds true, Dennard scaling [27] does not. Dennard scaling is the observation that as transistors get smaller, the power used by each transistor shrinks. Unfortunately, the shrinking factor (although it still holds true) is not fast enough to cope with the increase in the number of integrated transistors and as a result, the overall circuits’ power demands have been increased. Consequently, in the post-Dennard era contemporary ICs with billions of transistors are **underclocked** because of two reasons: a) to dissipate less power in order to extend the battery life of mobile devices and b) to dissipate the power without violating the power dissipation limits that result to overheating.

In the past the higher integration level from era to era was followed by an increase in the operational frequency and so the TAT per transistor was decreasing. As shown in Figure 10 the operational frequency of contemporary ICs tends to saturate the last years breaking the frequency prosperity. This seems an inevitable effect as long as the material limits have been reached and there are not any other material level technologies to fill this gap. Manufacturers no longer provide a processors’ power consumption characteristic but they provide the Thermal Design Power (TDP) which is the maximum amount of power that can be dissipated. Contemporary processors (like Ivy Bridge which is Intel’s 22nm series) exhibit TDP at the range of [35 - 130] Watts. The TDP limitation on the amount of power that a chip can dissipate introduced additional two testing obstacles:

- The underclocked circuits require more testing time compared to overclocked circuits. Although, more tests are needed to test higher density technologies compared to previous technologies, the tests cannot be conducted faster anymore.
- Traditional testing techniques decrease test cost by concurrently targeting as many



Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten  
Dotted line extrapolations by C. Moore

Figure 10: Moore’s Law in Respect to Transistors Number, Single Thread Performance, Frequency, Power and Number of Cores

defects as possible, leading thus to elevated test power consumption, which can be several times higher than that in functional mode [8]. The TDP limits forbid that because the tested devices might be harmed or tests may fail their purpose.

Power unaware testing techniques cause the circuit to consume much more power in test mode than in normal mode [10, 30, 44, 81, 85, 97, 138]. It was shown in [138] that test power can be more than twice the power consumed in normal functional mode. Specifically, some reasons for this gap between normal’s and test’s mode power consumption include:

- ATPG tools tend to generate test patterns with a high toggle rate in order to reduce pattern count and thus test application time. Therefore, the node switching activity of the device in test mode is often several times higher than that in normal mode.
- Parallel testing is often used to reduce test application time, particularly for testing MCSoc devices. This parallelism inevitably increases power dissipation during test.
- Circuitry inserted in the circuit to alleviate test issues is often idle during normal operation but may be intensively used in test mode. This surplus of active elements during test also induces an increase of power dissipation.
- Elevated test power can come from the lack of correlation between consecutive test patterns, while the correlation between successive functional input vectors applied to a given circuit during normal operation is generally very high [122].

As a result old test practices are deprecated and low power testing techniques are required. The new techniques ought to be faster than power-unaware testing techniques and on the same time to handle the power dissipation limits.

## 5.2 Multi-Core Systems-on-Chips and Intellectual Property Cores

The sustaining of Moore's law growth is essential not only because it offers prosperity on almost every aspect of human life but also because it provides payback of the huge capital investment of semiconductor's industry (an industry with starting capital that overpasses 3 billion dollars). To fill the processing gap of the no-longer-increased operating frequency, the industry has counter-proposed Multi-Core Systems-on-Chips (MCSOCs). They are based on exploiting the concurrent processing in order to offer faster systems.

However, MCSOCs require specialized assemble processes that increase test cost. There is a general agreement with the *rule of ten*, which says that the cost of detecting a faulty IC increases by an order of magnitude as we move through each stage of manufacturing, from device level to board level to system level and finally to system operation in-the-field. Nevertheless, MCSOCs brought not only challenges but also oportunities. Techniques such as parallel and multi-site testing [37] have been introduced. These techniques exploit capabilities of new generation DFT-aware test equipments [9, 52] for *test resources sharing*. In order for this technology to be efficient, DFT methodologies with *reduced pin-count interface* between ATE/CUT are required.

Intellectual Property (IP) cores that usually reside within MCSOCs complicate further testing. There are two main types of components within an MCSOC: the cores and the user defined logic (UDL). A core is a pre-designed, pre-verified silicon circuit block that can be used in building a larger or more complex application on a semiconductor chip. Cores can perform a wide range of functions (e.g., digital signal processors, RISC processors, or DRAMs) and can be found in a number of technologies (e.g., complementary metal-oxide-silicon (CMOS) logic, DRAM and analog circuits). Furthermore, the more complex cores come in hierarchical compositions (i.e., complex cores comprise a number of simple cores). Often these cores are products of technology, software, and know-how that are subject to patents and copyrights. Hence, a core block represents IP that the core builder licenses to the core user. Therefore, the core user is not always entitled to make changes to the core and is forced to reuse it as is (as a black box), being knowledgeable only about the cores functionality, however, not about the implementation details. In addition, while ICs are delivered to the customer in a manufactured and tested form, cores are delivered in a range of hardware description levels (soft, firm, and hard). These two fundamental differences influence not only the design of the MCSOCs, but also their testing.

Usually, IP cores are accompanied by pre-computed and pre-compacted test sets i.e. test sets with high fill rate. The compression efficiency of linear-based TRP methods drops drammatically when they are applied on test sets with high fill rate, because there are not many undefined values. On the other hand, although code-based compression methods are more efficient at compressing test sets with high fill rate, there are not any industry tools that supports them, because their compression efficiency on the test sets with low fill is moderate.

## 6 Test Response Partitioning Techniques

In order to offer high compression, TRP techniques usually exploit the following inherent properties of test cubes (test cubes are vectors consisting of ‘0’, ‘1’ and ‘x’ values):

- 1) The correlation between the specified ‘0’, ‘1’ values that stems from the structural correlation of faults [110],
- 2) The large amounts of unspecified (‘x’) values.

Code-based techniques exploit the correlations between the specified values, while linear-based techniques exploit the large amount of unspecified values.

The most widely adopted linear-based method is that of reseeding LFSRs [58, 60, 61]. LFSR reseeding exploits the low fill rate of test cubes. In [79] ring generators were proposed as an alternative to classical LFSRs and in [88] embedded deterministic test (EDT) was presented. Other well known techniques have been presented in [3, 8, 22, 23, 90, 109, 136, 137]. However linear-based methods do not exploit the high correlation between test cubes’ specified bits. In addition, they are ineffective for testing IP-cores which are usually accompanied by pre-computed and pre-compacted test sets. The main idea behind LFSR reseeding is to exploit the low density of specified bits in the test cubes (i.e., test patterns with ‘x’ logic values) in order to compress test cubes into LFSR seeds. A seed is computed by solving a system of linear equations, where the initial state of each LFSR cell is considered to be a binary variable. Although there are many LFSR reseeding techniques, each technique falls in one of the following categories: a) static reseeding or b) dynamic reseeding. In static LFSR reseeding the contents of the linear decompressor are flushed during reseeding, while in dynamic approaches they are not (flushed).

Many TRP techniques have been proposed that are suitable for cores of known structure [7, 38, 39, 50, 60, 75, 89, 111, 115, 131]. The high efficiency of these techniques is mainly attributed to the exploitation of the capabilities offered by the ATPG and fault simulation tools during the compression process. However, in the case of IP cores, where the structure of embedded cores is hidden from the system integrator, the utilization of such tools is not an option. The only option provided in these cases is to directly compress a pre-computed and usually pre-compacted test set which is provided by the core vendor. As a result, various methods have been proposed so far for compressing pre-computed test sets of IP cores. Among them, many methods utilize linear decompressors [3, 62, 63, 63, 65, 98, 117, 124] whereas others utilize various compression codes [14–16, 35, 45, 53–56, 82, 94, 106, 107, 120]. Although, these techniques are efficient for compressing pre-compacted test sets of IP cores, they are less efficient for cores of known structure. Also, there are also methods that do not belong in any of the above categories, e.g., [70] and [90]. Commercial tools have also been developed [5, 59, 88].

The next sections present, briefly, some of the most popular TRP techniques: the classical static LFSR reseeding, the window-based LFSR reseeding, the dynamic/partial LFSR reseeding, and the Optimal Selective Huffman (OSH) code-based technique.

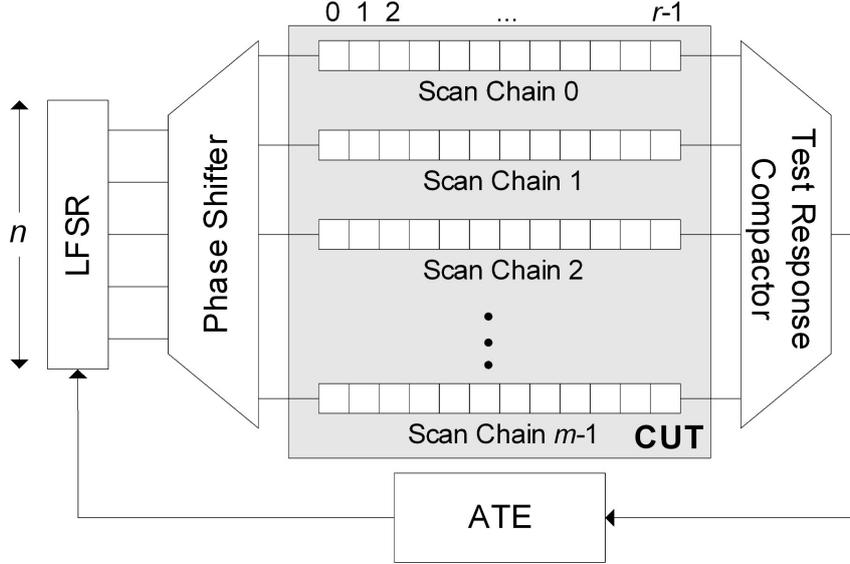


Figure 11: Classical LFSR-based decompression architecture

## 6.1 Static LFSR reseeding techniques

In static reseeding, test cubes are encoded into seeds, and every seed is loaded into a Linear Feedback Shift Register (LFSR) before decompression begins. Static reseeding, in its classical form, uses one new initial LFSR state (seed) for encoding a single test cube of the test set [58]. The major drawback of this approach is that it offers limited compression. Many other static LFSR reseeding methods have been proposed in the past [45, 51, 62, 63, 75, 108, 117] which offer better compression than [58]. A particularly efficient approach is window-based reseeding [51], where each seed is used to generate more than one test vector i.e., each seed is expanded into a window of test vectors.

### 6.1.1 Classical LFSR Reseeding

The classical LFSR reseeding scheme [58] uses an LFSR-based decompression logic as presented in Figure 11. Every  $n$ -bit seed ( $n$  is the LFSR size) is transferred from the ATE to the LFSR, where it is expanded into a test vector of  $m \times r$  bits ( $m$  is the scan-chain volume and  $r$  the scan-chain length) and is loaded into the scan chains. When a test vector is loaded into the scan chains of the CUT, the response of the previous vector is shifted out to the Test Response Compactor. The phase shifter [4, 43, 73] is used to reduce the linear dependencies [80] of the bit sequences generated by the LFSR cells. Algorithms for designing synthesizable phase shifters are presented in [79, 86, 87]. The test set of the core consists of test cubes of  $m \times r$ -bits each, and every one of them is compressed into an  $n$ -bit seed ( $n \ll m \times r$ ) which is calculated by solving a system of linear equations. This system is formed according to the specified bits of the test cube [58] (the ‘X’ bits are filled with pseudorandom data during decompression). Specifically, the initial state of the LFSR is considered as a set of binary variables  $\alpha_0, \alpha_1, \dots, \alpha_{n-1}$ . At every clock cycle,  $m$  linear expressions of these variables are generated at the  $m$  outputs of the phase shifter.

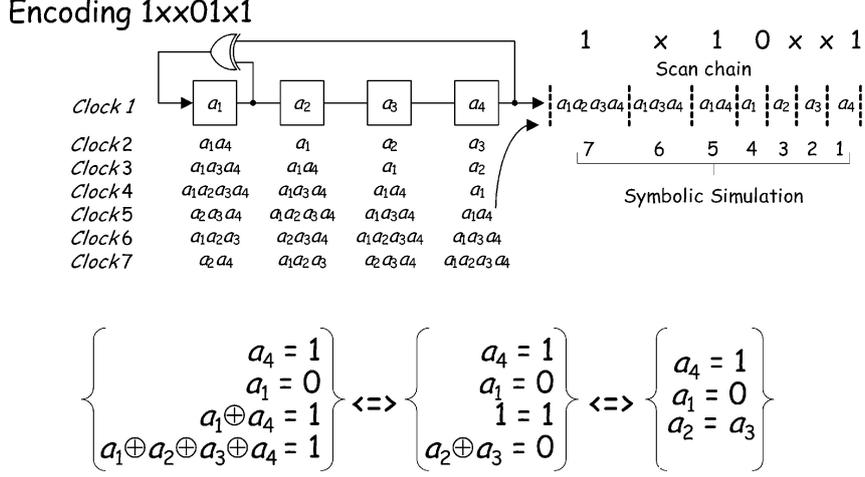


Figure 12: Classical LFSR reseeding example

During  $r$  successive clock cycles,  $m \times r$  linear expressions are generated at the outputs of the phase shifter, and each one of them corresponds to one of the  $m \times r$  scan cells. Thus, each bit of a test cube corresponds to exactly one linear expression. Every linear expression corresponding to a specified bit of a test cube is set equal to that bit, and in this way the system of linear equations is formed (the unspecified bits of the test cubes are not considered during this step). The solution of this system is the seed of the LFSR. The system with the maximum number of linear equations corresponds to the test cube with the maximum number of specified bits,  $s_{max}$ , which in turn determines the minimum required LFSR size. As it was shown in [58], if the LFSR size  $n$  is equal to  $s_{max} + 20$ , then the probability of not being able to solve the linear system for encoding a test cube is less than  $10^{-6}$ . However, LFSR polynomials with size less than  $s_{max} + 20$  exist, which can compress all test cubes [12].

*Example 1.* Figure 12 presents a reseeding example. On the upper left corner of the figure there is the test cube to be encoded, while under it there is the utilized LFSR. At the left of the LFSR there are the clocks numbered and the symbolic states of the LFSR, presented line-by-line for each cycle though symbolic simulation (the terms  $\alpha_i \alpha_j \dots \alpha_k$  at the symbolic simulation are used to denote the values  $\alpha_1 \oplus \alpha_j \oplus \dots \oplus \alpha_k$ , where  $\oplus$  is the XOR logic function). Suppose that a scan chain is directly loaded with the contents of the last cell of the LFSR. The contents of the scan chain for each cycle can be seen at the right of the LFSR. By applying the test cube on this symbolic representation of the scan chain's contents and equalizing its defined bits with the symbolic representations, we form the linear system. The solution of this system is the LFSR's seed that generates the encoded test cube. Notice that the number of equations that form the linear systems strongly depends to the number of specified bits of the test cube. ■

### 6.1.2 Window-Based LFSR Reseeding

According to the classical LFSR reseeding, every seed is used for encoding a single test cube. The achieved compression in this case is moderate, since usually in a test set there are many test cubes with fewer specified bits than the bits of the maximum specified test cube. As a result, a lot of variables remain unspecified when the corresponding systems are solved, and therefore much of the potential of LFSR's encoding is wasted.

Various methods have been proposed for better utilization of LFSR's variables, [60, 88, 117, 135] to name a few. A very attractive one is to utilize the same seed for encoding more than one test cube in a sequence of  $L$  pseudorandom vectors. In other words, each seed is expanded into a window of  $L$  vectors, instead of one. The number of test cubes encoded in the window is usually much smaller than  $L$ , which means that useless vectors are also applied to the CUT. This approach is very effective since for every test cube,  $L$  (and not just one) systems of equations are constructed, and among the solvable systems, the one resulting in the highest compression is selected. In other words, each test cube is encoded in such a way so as to maximize the overall encoding efficiency. There are many ways to encode multiple test cubes in an  $L$ -vector window. One very effective algorithm for minimizing the number of seeds is the following [25, 49]: initially, the test cube with the highest number of specified bits is selected and the system corresponding to the first vector of the window is solved (the selection of the LFSR polynomial and the phase shifter guarantees that this system is always solvable). The remaining test cubes are selected iteratively according to the following criteria:

- Among the solvable systems that correspond to the test cubes containing the maximum number of specified bits, we identify those that their solution leads to the replacement of the fewest variables in the  $L$ -vector window.
- Among them, we find those corresponding to the cube that can be encoded the fewest times in the window.
- Finally, among them we select the system nearest to the first vector of the window.

After solving the selected system, some of the variables are replaced by logic values, whereas the rest remain unspecified and they are utilized for encoding additional test cubes. The construction of a seed is completed when no system for any of the unencoded test cubes can be solved in the  $L$ -vector window. Although, test set embedding techniques, such as window-based LFSR reseeding, can achieve high compression efficiency they suffer from long test sequences.

## 6.2 Dynamic LFSR Reseeding

Dynamic reseeding methods [59, 60, 88] constitute another class of methods that offer high compression. In these approaches the content of the linear decompressor are not flushed during the reseeding and as a result any remaining unsolved variables inside the decompressor can be still exploited for compression.

As we mentioned for classical LFSR reseeding in Section 6.1.1, and we highlight again

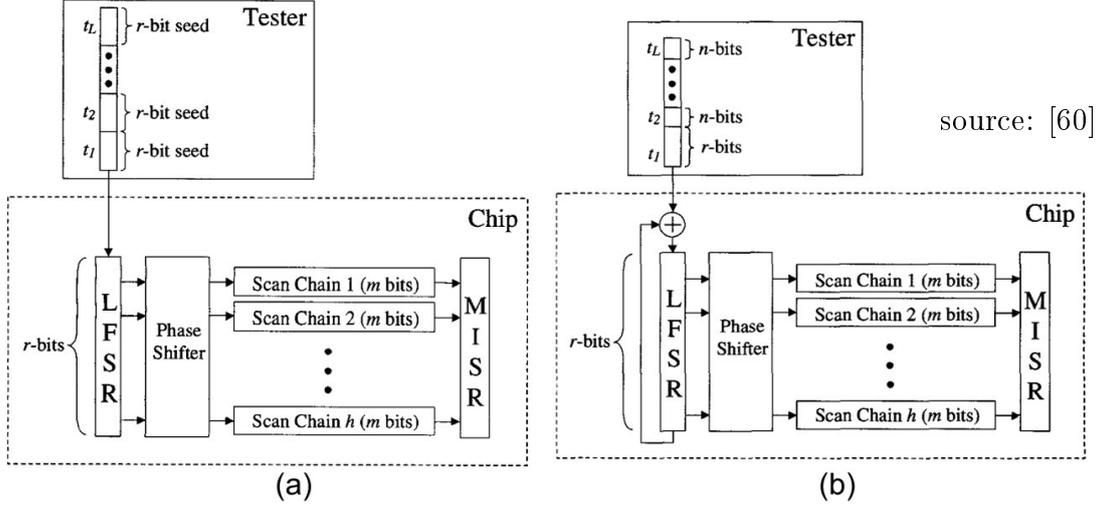


Figure 13: (a) Static reseeding versus (b) dynamic reseeding

in Figure 13a, an  $r$ -bit LFSR is loaded with an  $r$ -bit seed and then generates the desired test vectors. Afterwards, it flushes its contents and it is being loaded with a new  $r$ -bit seed etc. This kind of reseeding results into wasted variables because the size of the LFSR  $r$  depends on the number of bits  $s_{max}$  of the most specified test cube in a test set. Dynamic LFSR reseeding is shown in Figure 13(b). Note that an extra XOR gate is included in the feedback of the LFSR. The LFSR of the figure has only one input and loads serially the seeds on it. The initial  $r$ -bit seed it used to initialize the LFSR and it is let to operate and generate the generated test vector. Afterwards, instead of flushing its contents, it dynamically (the term *dynamic* used by [60] to denote “without flushing”) loads the next  $n$ -bit seed (where  $n < r$ ; from this property stems the term *partial* LFSR reseeding and it was first introduced by [135] with the term *variable-length seeds*) without flushing. As a result any unresolved variables from the previous  $r$ -bit seed remain active in the LFSR and may be utilized in a later phase. The next generated vector can now exploit any unresolved variables from previous seeds together with the newly inserted  $n$ -bit seed.

The contents of an LFSR that is dynamically and partially reseeded may be symbolically simulated in a similar way with the symbolic simulation of static LFSR reseeding. An example of the new symbolic simulation procedure and linear systems forming is illustrated in Figure 14. However, there is a difference between the symbolic simulation of static and dynamic reseeding. The symbolic simulation for dynamic reseeding is a very time consuming process because the linear equations for the test cubes need to be solved altogether [60]. As a result, dynamic reseeding may not be scalable to large designs unless proper actions are taken. To this end [60] proposed a test set partitioning method which provides sub-optimal results but reduces the CPU time of dynamic reseeding symbolic simulation and linear equations solving. The reported CPU times at [60] are in the order of hours (on CPUs of that time) when the partitions consist of hundreds of test cubes. In the experiments presented in this dissertation and concern dynamic LFSR reseeding we

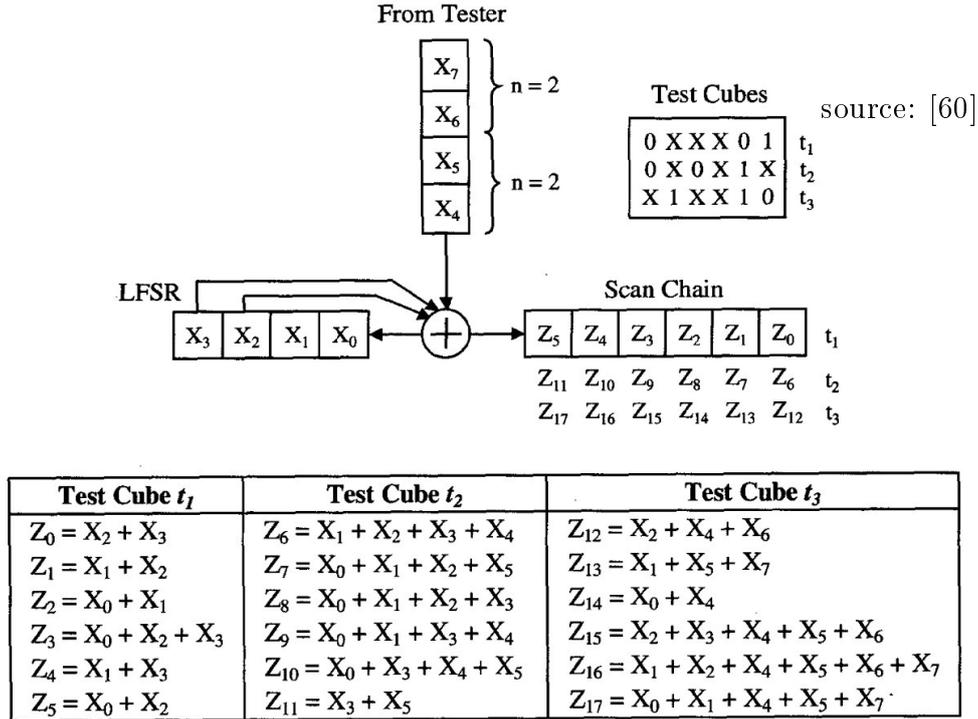


Figure 14: Partial/Dynamic LFSR symbolic simulation

have not implemented this partitioning technique as we intended to provide the most favorable results in terms of compression for dynamic reseeding. Nevertheless, even for our largest benchmark circuit, “Ethernet” with 10 thousands of test cubes from the IWLS [1] benchmarks suite, the CPU run-times for forming and solving the equations of dynamic LFSR reseeding without the partitioning technique of [60] are in the order of hours (on contemporary CPUs).

### 6.3 Code-based Techniques

Code-based schemes use data compression codes to encode the test cubes. This involves partitioning the original data into symbols, and then replacing each symbol with a code word to form the compressed data. To perform decompression, a decoder simply converts each code word in the compressed data back into the corresponding symbol. Code-based compression techniques are classified depending on whether the symbols have a fixed or variable size (symbols have the same or different numbers of bits respectively) and whether the codewords have a fixed or variable size. Therefore, four categories follow: fixed-to-fixed [91, 133], fixed-to-variable [6, 45, 46, 54, 69, 72, 99, 134], variable-to-fixed [47, 123, 132] and variable-to-variable [12, 14–16, 53, 55, 56, 67, 82, 106, 107].

The first data compression codes that researchers investigated for compressing scan vectors encoded runs of repeated values. In [46, 47] a scheme based on run-length codes that encoded runs of repeated ‘0’ values using fixed-length code words is proposed. In

Table 1: Test Set Partitioned to Data Blocks and Distinct Blocks' Frequencies

Test Set $T$	Distinct Blocks	Occur. Freq.
1010 0000 1010 1111	1010	9/20
1111 0000 1010 0001	0000	5/20
1010 0000 0010 1010	1111	3/20
0000 1010 1010 0000	0001	2/20
1010 1111 1010 0001	0010	1/20

[14] a technique based on Golomb codes that encodes repeated values with variable-length codewords is presented. The use of variable-length code words allows efficient encoding of longer runs, although it requires a synchronization mechanism between the tester and the chip. Further optimization is achievable by using frequency-directed run-length (FDR) codes [15, 16, 29] and variable-input Huffman codes [35, 45, 53, 54, 56], which customize the code based on the distribution of different run lengths in the data. Other techniques that utilize other compression codes or multiple codes simultaneously are [6, 82, 106, 107, 132].

Code-based schemes are very effective in exploiting correlations in test cubes and they do not depend on the Automatic Test Pattern Generation (ATPG) process used. Consequently, they are very effective on pre-computed (and usually pre-compacted and densely specified) test sets for Intellectual Property (IP) cores. However, they suffer from several serious drawbacks that prohibit their use in industrial designs: they do not exploit the low fill rate of test cubes; they impose long testing times as they cannot exploit the large number of scan chains; they require extensive interaction with the tester.

### 6.3.1 Optimal Selective Huffman

The Huffman code [42] is a fixed-to-variable code that uses short codewords to encode frequently occurring blocks and long codewords for the less-frequent ones. The Optimal Selective Huffman (OSH) code encodes only the  $m$  most frequent blocks [45] while the rest of the blocks remain un-encoded and they are distinguished by using an extra Huffman codeword [54].

Let us assume a core with  $n$  scan chains of length  $r$  (we assume a balanced scan structure where the shorter scan chains are padded with 'x' logic values). Each scan slice constitutes a single block. Let  $T$  be a set of test cubes and  $|T|$  be its size in bits.  $T$  is partitioned into  $|T|/l$  data blocks of size  $l$ . Among these blocks the  $m$  most frequent distinct blocks,  $b_1, b_2, \dots, b_m$ , with frequencies (probabilities) of occurrence  $f_1 \geq f_2 \geq \dots \geq f_m$  respectively are stored in a dictionary and they are encoded using  $m$  Huffman codewords. The rest of the blocks with an aggregate frequency  $f_{un} = f_{m+1} + f_{m+2} + \dots$  remain not-encoded and a single codeword is used to precede them (they are stored in a raw form in the compressed test data [54]). A binary tree is constructed beginning from the leaves and moving towards the root. For every dictionary entry  $b_i$  a leaf node is

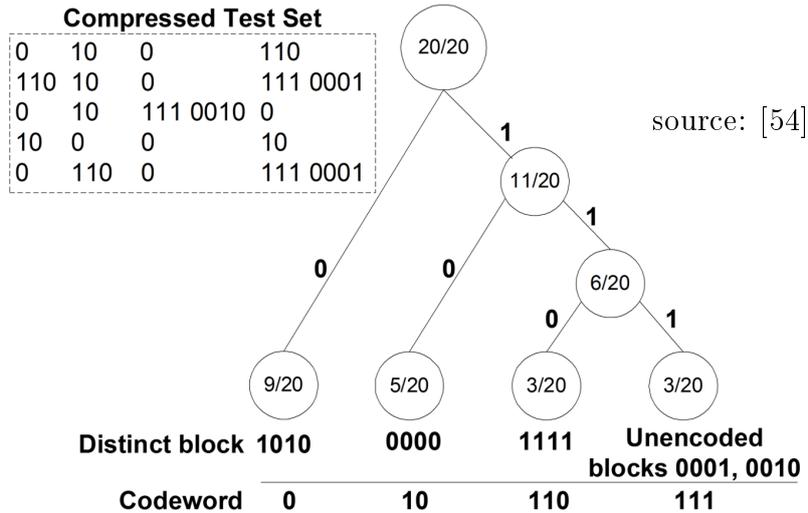


Figure 15: Optimal Selective Huffman Example

generated, and a weight equal to  $f_i$  is assigned to it. The pair of nodes with the smallest weights is selected first and a parent node is generated with a weight equal to the sum of the weights of both nodes. This is repeated iteratively, until the root is left unselected (each node can be selected only once). After the tree is constructed each leaf node is assigned a codeword as follows: starting from the root, all nodes are visited once and the logic ‘0’ (‘1’) value is assigned to each left (right)-child edge. The codeword of block  $b_i$  is the sequence of the logic values of the edges on the path from the root to the leaf node corresponding to  $b_i$ .

*Example 2.* Consider the test set of Table 1 and that  $m = 3$ , that is 0001 and 0010 are the unencoded blocks. The sum of the occurrence frequencies of 0001 and 0010 is equal to  $2/20 + 1/20 = 3/20$ . The OSH encoding as well as the compressed test set are given in Figure 15. The encoding distinct blocks 1010, 0000 and 1111 are encoded by codewords 0, 10 and 110 respectively. The unencoded data blocks are distinguished by the the 3-bit codeword 111. Finally, the number of bits for the compressed test data is 42 bits (from the nested Table of Figure 15), while the uncompressed test data were 80 bits (from Table 1). ■

Despite the fact that there are many blocks in a test set consisting mostly (or even entirely) of ‘x’ values, each and every one of them has to be encoded using a separate codeword. As a result even if a test was fully specified still many bits would be required for its encoding. Assume that the test set of Table 1 was fully unspecified. Then 20 bits (1 bit per block) would be the size of the compressed test set by the OSH method. It becomes obvious that although, the selective Huffman code [45], [54] offers low cost decompressors and high compression at the same time, it can not be used for industrial applications because it cannot exploit the unspecified values in the test sets. Another important drawback is that it requires a synchronization mechanism between the ATE and the CUT.

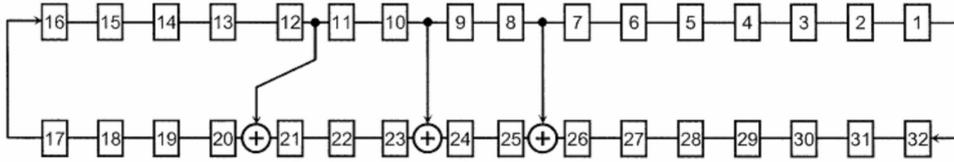


Figure 16: A Ring Generator

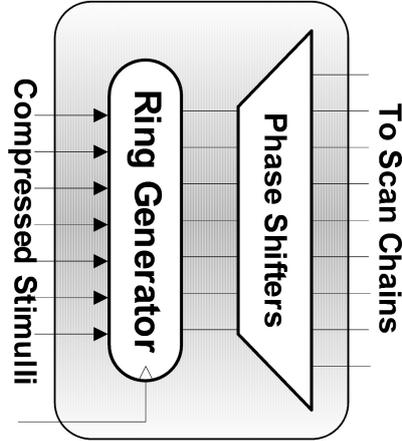


Figure 17: EDT basic Architecture

## 6.4 Industry Practice: Embedded Deterministic Test (EDT)

Embedded Deterministic Test (EDT) was proposed in [88] and it is constantly being enriched with new properties since then. So, it is a collection of tools and methods to create a successful embedded testing architecture based on a modified LFSR called *ring generator*.

Similar to linear-based approaches, ring generators are based on prime LFSR polynomials. Usually (if not always), prime polynomials XOR taps synthesis result to high fan-outs of the decompressors and as a result slow decompression feedback operation. In [79] a transformation method was presented of an LFSR to a more synthesizable-friendly form with XOR's fan-out maximum value of 2. In Figure 16 a ring generator is presented. Figure 17 illustrates the basic EDT architecture:

- Compressed data are provided to the ring generator [79] from the ATE.
- The pseudorandom test sequences generated by the ring generator are shifted by the Phase Shifters [4, 43, 73, 79, 86, 87] and then fill the scan chains.

Every generated vector of length  $L$  is loaded into the scan chains as  $K$  slices of  $S$  size each where  $S$  is also the number of the scan chains and  $L = K \times S$ .

EDT utilizes dynamic/partial reseeding on ring generators. Symbolic simulation for dynamic reseeding requires all the variables to be handled together (see Section 6.2). This is a bottleneck for the execution time and it was handled by [60] with a partitioning of the test set. But, the original algorithm of EDT proposed in [88] is not applied on a pre-computed test set, so this partitioning is not feasible. So, EDT has adopted a variable's

elimination strategy to handle this bottleneck. Moreover, variable elimination exploits contemporary ATE's REPEAT command [118]. Suppose that variables are injected from a channel between the ATE and the ring generator. Any unresolved variable is decided if it will be eliminated (the elimination is to be set to '1' or '0' for symbolic simulation scaling reasons) or not based on some criteria (Variables Elimination criteria). Some, criteria are based on profiles on the number of specified bits of the test cubes (Non-Adaptive Variables Elimination) or ad-hoc criteria during the compression based on the remaining free variables (Adaptive Variables Elimination). In order not to compromise the compression by this approach, ATE's REPEAT command is exploited and the eliminated variables are set to the previous value that was injected from the same channel. This way the compression tool of EDT overpasses the bottleneck of handling all the variables together and also becomes applicable in synergy with ATPG and fault simulation.

EDT uses fault simulation after the generation of a test pattern in order to drop any easy-to-detect faults (faults with test cubes that have few specified values and are randomly tested). The ATPG generates test cubes during the compression (as a result the compression algorithms gets the next-to-compress test cube directly from the ATPG tool). This interaction between ATPG/compression-tool can maximize compression, especially for  $N$ -detection test sets, because faults are directly dropped during the fault simulation step and they are not considered from the ATPG tool for the generation of the next test cube.

## 7 Low-Power Testing Techniques

A drawback of both linear-based and symbol-based test data compression techniques is that they elevate switching activity beyond acceptable levels and thus degrade production yield [101]. Power consumption during scan testing consists of two switching activity components, namely *shift* and *capture* power. In particular, shift switching activity (referred also as shift power) is caused during the shift (scan in-out) mode of scan designs when successive complementary logic values are shifted into the scan chains. When complementary values are shifted into a scan chain they generate transitions on the scan cells while they travel to their final destination. The transitions on the scan cells, inevitably, generate more transitions at the combinational part of the circuit that is attached at the scan chain. The result is increased switching activity during the shift in-out mode. This increased switching activity is responsible for the average power consumption that increases the generated heat during testing beyond the acceptable TDP limits. On the other hand, capture switching activity (referred also as capture power) is caused during the capturing of the responses on the scan cells. The transitions generated during capture mode may increase the instantaneous power demand of the CUT leading to ground/voltage bounces that introduce noise at CUTs signal values. Capture power values above certain limits can undermine the reliability of the testing procedure causing yield loss of operational CUTs that appear mistakenly as faulty. To alleviate switching activity during scan testing

various techniques have been developed.

Numerous methods have been proposed in the literature for limiting power consumption during testing, targeting shift power [11, 12, 20, 24, 31, 48, 65, 78] or capture power [17, 71, 93, 125–129]. In addition, some methods simultaneously target the reduction of both shift and capture switching activity [13, 57, 68, 92, 96]. These methods can be further categorized as being either structural [12, 17, 20, 24, 31, 57, 65, 78] or algorithmic [96, 125, 126] based on their nature. Structural methods interfere with the scan design architecture by modifying it for low power purposes. On the other hand at the algorithmic methods there are low power ATPG techniques and test cubes manipulation techniques [11, 13, 48, 68, 71, 92, 93, 125, 128, 129] also known as X-filling.

Below the most known structural and algorithmic low power testing techniques are briefly presented.

## 7.1 Structural Low-Power Testing Approaches

Even though traditional TDC techniques (like for example [3, 8, 45, 54, 58, 61, 88]) are very efficient in compressing test data, they become deprecated under power dissipation limitations. Especially large power demands exhibit the linear decompressors, because they fill the ‘X’ values pseudorandomly and they increase thus both the shift and capture power during scan testing. Specifically, linear decompressors are very effective in compressing the test data, they elevate the power dissipation during testing above the functional power budget of the circuit. A few symbol-based TDC techniques such as [14–16, 46, 82], inherently offer low shift power but they are not suitable for cores with multiple scan chains.

To comply with power consumption requirements, linear decompressors which offer low switching activity during testing have emerged [21, 24, 64, 78]. These techniques require additional data to control the switching activity. Specifically, the state-of-the-art low power dynamic reseeding [18, 78] utilizes a shadow register to offer low power shift testing by repeating test data but it requires additional test data compared to EDT [88] for controlling the low power operation of the decompressor. In [19] selective scan enable deactivation is used for low capture power and in [112] presents a TDC technique with narrow ATE-bandwidth requirements. The method proposed in [23] exploits similarities between test cubes to offer higher compression and utilizes both shadow registers and scan enable deactivation to generate low power vectors.

### 7.1.1 Low-power Linear Decompressors

Figure 18 presents the classical scan based architecture. The CUT consists of  $c$  scan chains of length  $r$  (for simplicity we assume that all scan chains are of equal length). The compressed test data are downloaded from the ATE, they are decompressed using the embedded decompressor and they are shifted into the scan chains. For applying a test vector to the CUT the decompressor first generates  $r$  successive test slices of size  $c$  which

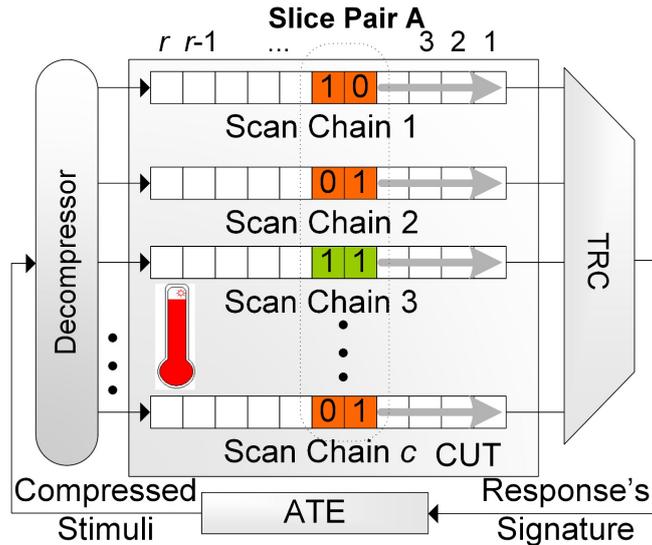


Figure 18: Switching Activity caused by Successive Slices

are shifted into the scan chains to reach their respective scan slices (hereafter, the term test slice  $t_j$  refers to the test bits of test cube  $t$  which correspond to scan slice  $j$  with  $j \in [1, r]$ ). After the last test slice of  $t$  (i.e.  $t_r$ ) is shifted into the scan chains,  $t$  is applied to the CUT and the response is shifted out concurrently with the loading of the next test vector. Linear decompressors fill ‘X’ values pseudorandomly, and thus they fail to control the number of incompatibilities between successive test slices.

In Figure 18, every pair of successive test slices exhibits potential bitwise incompatibilities, i.e. pairs of successive complementary test bits loaded into the same scan chains. For example test slices denoted as “Slice Pair A” in Figure 18 are incompatible in the bit positions corresponding to scan chains 1, 2,  $c$ . As the test slices travel through the scan chains during the scan-in process, every pair of complementary successive test bits causes transitions on the scan chains which propagate through the combinational logic and cause switching activity to the CUT. The number of incompatibilities between successive test slices can be reduced by exploiting the unspecified values which exist in large volumes in test sets. However, linear decompressors fill ‘X’ values pseudorandomly, and thus they fail to control the number of incompatibilities between successive test slices.

### 7.1.2 Low-power EDT

The authors of [78] proposed a linear based encoding method which exploits the ‘X’ values, wherever they exist, to reduce incompatibilities between successive test slices, and thus to reduce shift power. According to this method, whenever a group of  $k$  ( $k > 1$ ) successive test slices of a test cube are compatible (i.e., every slice in this group exhibits no bitwise incompatibilities with any other slice in this group) one test slice  $S_k$  is computed which is compatible with all  $k$  test slices. This slice is encoded using the ring generator and it is loaded into the scan chains for  $k$  successive clock cycles. This is achieved by the use of a

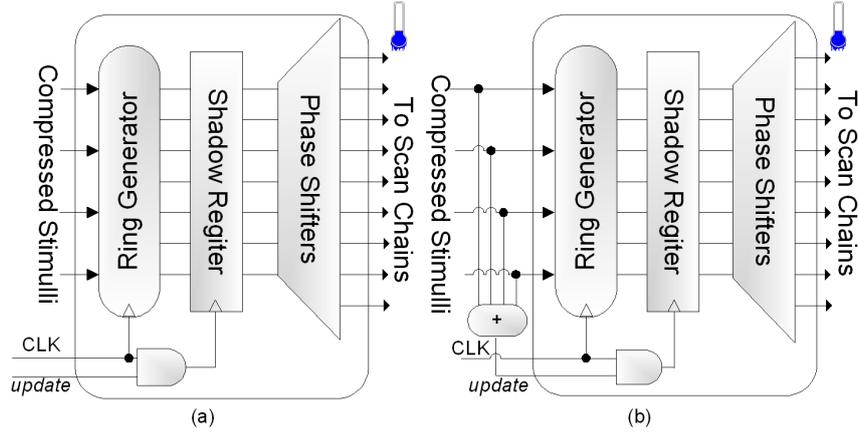


Figure 19: (a) Low power EDT controlled by an additional “update” channel, (b) Low power EDT controlled by compressed stimuli

shadow register shown in Figure 19 which can hold its contents if it is properly controlled. Specifically, instead of generating the first slice of this group, the ring generator generates slice  $S_k$  and it transfers this slice to the shadow register. This is called UPDATE operation. During the next  $k$  successive clock cycles, the shadow register holds its contents and loads the scan chains with slice  $S_k$ . This is called HOLD operation. The selection between these two operations of the shadow register requires additional control data which are either provided directly from the ATE (Figure 19a) or they are encoded as compressed stimuli (Figure 19b). In both cases the additional cost is considerable especially when the number of ATE channels is small and the number of slices per vector is large.

## 7.2 Algorithmic Approaches: Low-Power X-Filling Techniques

X-filling techniques aim at a power-aware logic assignment of the unspecified X-bits. X-filling has negligible impact on ATPG process, and affects neither the scan chain structure nor the circuit under test (CUT). Moreover, they can be combined with other techniques for further reducing test power.

A popular X-filling method for reducing shift power is Fill-Adjacent (FA) technique [11]. This technique targets only the scan-in portion of the shift power, but it also reduces the scan-out power, because, as shown in [13], the scan-in power is highly correlated to the scan-out power. In addition, it can be easily combined with capture-power reduction techniques such as the Preferred-Fill (PF) technique [92, 93], to provide an efficient unified power-reduction solution. The FA and the PF X-filling techniques, for shift and capture power reduction respectively, are briefly presented below.

### 7.2.1 Overview of Fill-Adjacent

A well-known technique to reduce shift power dissipation is the Fill-Adjacent (FA) technique [11]. This technique targets only the scan-in portion of the shift power. The

Table 2: Fill-Adjacent X-Filling\*

	Test Cube Block	FA
i	0x...x0, 0x...x, x...x0	00...00
ii	1x...x1, 1x...x, x...x1	11...11
iii	0xx...x1	011...11
iv	1xx...x0	100...00

\*the rightmost bit is loaded first into the scan chain

simplicity of FA and the reason that it can reduce the overall shift power (both scan-in and scan-out, as shown in [13]) is the key of its success.

Every two complementary consecutive test bits loaded into a scan chain generate switching activity as they travel along the scan chain. The FA technique minimizes the shift power by exploiting the X-bits of the test cubes in order to minimize the volume of the consecutive complementary test bits loaded into the scan chains as well as the distance they travel along the scan chains. For instance, consider a CUT with  $c$  scan chains, and assume that the test cube segment  $S_j = XXX1XX01XX0XXX1$  has to be loaded into scan chain  $j$  ( $1 \leq j \leq c$ ) from right to left. By applying FA to fill the Xs, we get the test vector segment  $T_j = 1111000010001111$ . Table 2 shows all possible X-fillings produced by the FA technique. The first column shows all possible blocks of test bits comprising any test cube segment that consists of  $n$  ( $n \geq 1$ ) unspecified logic values bounded at the left and/or right by specified logic values. The second column shows the X-filling produced for all these blocks.

### 7.2.2 Overview of Preferred-Fill Techniques

The Preferred Fill (PF) technique (denoted hereafter as PF) is an X-filling technique for reducing the switching activity during capture [16, 17]. Consider a two-pattern Launch-On-Capture (LOC) test  $\langle V_1, V_2 \rangle$  where  $V_1 = (v_{11}, v_{12}, v_{13}, \dots, v_{1n})$  is the first  $n$ -bit vector applied on the CUT and  $V_2 = (v_{21}, v_{22}, v_{23}, \dots, v_{2n})$  is the response of  $V_1$  which is applied as the second test vector to the CUT. If the logic value of  $V_1$  corresponding to cell  $i$ , (i.e.,  $v_{1i}$ ) is unspecified then it should be filled with value 1(0) provided that the probability of  $v_{2i}$  (i.e., the logic value of  $V_2$  corresponding to the scan cell  $i$ ) taking the value 1(0) is higher than taking the value 0(1). In other words, the  $v_{1i}$  bit is filled with a value that is more likely to be held after the capture in the  $i^{th}$  scan cell.

### 7.2.3 X-Filling Limitations

A major drawback of power-aware X-filling techniques is that they are often accompanied by a reduction in defect coverage, since the impact on unmodeled fault coverage is not considered during X-filling. ATPG engines, on the other hand, increase the fortuitous detection of modeled as well as of unmodeled faults by filling randomly the Xs. However, this step elevates the test power.

## 8 Conclusions

The wide spreading of Very Deep Sub-Micron (VDSM) Integrated Circuits' (ICs), the architectural advancements that made possible the construction of Multi-core Systems-on-Chips (MCSoCs), and the power dissipation limitations imposed by the post-Dennard era created an explosive mixture for the upcoming manufacturing testing technologies. Failure in sustaining manufacturing testing cost low can make these advancements collapse.

## Acknowledgement

*This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) – Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.*



**European Union**  
European Social Fund



OPERATIONAL PROGRAMME  
EDUCATION AND LIFELONG LEARNING  
MINISTRY OF EDUCATION & RELIGIOUS AFFAIRS  
MANAGING AUTHORITY



EUROPEAN SOCIAL FUND

**Co- financed by Greece and the European Union**

## References

- [1] IWLS'05 circts., online: <http://www.iwls.org/iwls2005/benchmarks.html>.
- [2] “Test and test equipment,” in *International Technology Roadmap of Semiconductors*, 2007.
- [3] K. J. Balakrishnan and N. A. Touba, “Improving linear test data compression,” *IEEE Trans. Comput.-Aided Des.*, vol. 14, no. 11, pp. 1227–1237, 2006.
- [4] P. Bardell, “Design considerations for parallel pseudorandom pattern generators,” *Journal of Electronic Testing*, vol. 1, no. 1, pp. 73–87, 1990. [Online]. Available: <http://dx.doi.org/10.1007/BF00134016>
- [5] C. Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, B. Keller, and B. Koene-mann, “Opmisr: the foundation for compressed atpg vectors,” in *Test Conference, 2001. Proceedings. International*, 2001, pp. 748–757.
- [6] K. Basu and P. Mishra, “Test data compression using efficient bitmask and dictionary selection methods,” *IEEE Trans. Very Large Scale Integr.*, vol. 18, no. 9, pp. 1277–1286, Sep. 2010.
- [7] I. Bayraktaroglu and A. Orailoglu, “Concurrent application of compaction and compression for test time and data volume reduction in scan designs,” *Computers, IEEE Transactions on*, vol. 52, no. 11, pp. 1480–1489, 2003.
- [8] —, “Test volume and application time reduction through scan chain concealment,” in *Proc. DAC*, 2001, pp. 151–155.
- [9] J. Bedsole, R. Raina, A. Crouch, and M. S. Abadir, “Very low cost testers: Opportunities and challenges,” *IEEE Des. Test*, vol. 18, no. 5, pp. 60–69, Sep. 2001. [Online]. Available: <http://dx.doi.org/10.1109/54.953273>
- [10] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital Memory and Mixed-Signal VLSI Circuits*. Kluwer Academic Publishers, 2000.
- [11] K. Butler, J. Saxena, A. Jain, T. Fryars, J. Lewis, and G. Hetherington, “Minimizing power consumption in scan testing: pattern generation and dft techniques,” in *Test Conference, 2004. Proceedings. ITC 2004. International*, 2004, pp. 355–364.
- [12] A. Chandra and K. Chakrabarty, “Low-power scan testing and test data compression for system-on-a-chip,” *IEEE Trans. on CAD*, vol. 21, no. 5, pp. 597–604, may 2002.
- [13] A. Chandra and R. Kapur, “Bounded adjacent fill for low capture power scan testing,” in *Proc. VTS*, 2008, pp. 131–138.

- [14] A. Chandra and K. Chakrabarty, "System-on-a-chip test-data compression and decompression architectures based on golomb codes," *IEEE Trans. Comput.-Aided Des.*, pp. 355–368, 2001.
- [15] —, "Test data compression and test resource partitioning for system-on-a-chip using frequency-directed run-length (fdr) codes," *IEEE Trans. Comput.*, vol. 52, no. 8, pp. 1076–1088, Aug. 2003.
- [16] —, "A unified approach to reduce soc test data volume, scan power and testing time," *IEEE Trans. Comput.-Aided Des.*, vol. 22, no. 3, pp. 352–363, Mar. 2003.
- [17] B.-H. Chen, W.-C. Kao, B.-C. Bai, S.-T. Shen, and J. Li, "Response inversion scan cell (risc): A peak capture power reduction technique," in *Asian Test Symposium, 2007. ATS '07. 16th*, 2007, pp. 425–432.
- [18] D. Czysz, G. Mrugalski, N. Mukherjee, J. Rajski, P. Szczerbicki, and J. Tyszer, "Low power compression of incompatible test cubes," in *Proc. ITC*, 2010, pp. 1–10.
- [19] —, "Deterministic clustering of incompatible test cubes for higher power-aware edt compression," *IEEE Trans. Comput.-Aided Des.*, vol. 30, no. 8, pp. 1225–1238, Aug. 2011.
- [20] D. Czysz, G. Mrugalski, J. Rajski, and J. Tyszer, "Low power embedded deterministic test," in *VLSI Test Symposium, 2007. 25th IEEE*, 2007, pp. 75–83.
- [21] D. Czysz, M. Kassab, X. Lin, G. Mrugalski, J. Rajski, and J. Tyszer, "Low-power scan operation in test compression environment," *IEEE Trans. Comput.-Aided Des.*, vol. 28, no. 11, pp. 1742–1755, 2009.
- [22] D. Czysz, G. Mrugalski, N. Mukherjee, J. Rajski, and J. Tyszer, "On compaction utilizing inter and intra-correlation of unknown states," *IEEE Trans. Comput.-Aided Des.*, vol. 29, no. 1, pp. 117–126, 2010.
- [23] D. Czysz, G. Mrugalski, N. Mukherjee, and J. R. J. Tyszer, "Compression based on deterministic test vector clustering of incompatible test cubes," in *Proc. ITC*, 2009, pp. 1–10.
- [24] D. Czysz, G. Mrugalski, J. Rajski, and J. Tyszer, "Low-power test data application in edt environment through decompressor freeze," *IEEE Trans. Comput.-Aided Des.*, vol. 27, no. 7, pp. 1278–1290, Jul. 2008.
- [25] E. D. Kaseridis, Kalligeros, X. Kavousianos, and D. Nikolos, "Efficient multiphase test set embedding for scan-based testing," in *Inf. Pap. Dig. ETS*, 2005, pp. 147–150.

- [26] S. DasGupta, P. Goel, R. G. Walther, and T. W. Williams, "A variation of lssd and its implications on design and test pattern generation in vlsi," in *ITC*, 1982, pp. 63–66.
- [27] R. Dennard, "Design of ion-implanted mosfets with very small physical dimensions," *IEEE Journal of Solid State Circuits*, vol. SC-9, no. 5, pp. 256–268, April 1974.
- [28] E. B. Eichelberger and T. W. Williams, "A logic design structure for lsi testability," in *Proceedings of the 14th Design Automation Conference*, ser. DAC '77. Piscataway, NJ, USA: IEEE Press, 1977, pp. 462–468. [Online]. Available: <http://dl.acm.org/citation.cfm?id=800262.809170>
- [29] A. H. El-Maleh and R. H. Al-Abaji, "Extended frequency-directed run-length code with improved application to system-on-a-chip test data compression," in *Proc. ICECS*, 2002, pp. 449–452.
- [30] P. Girard, "Low power testing of vlsi circuits: problems and solutions," in *Quality Electronic Design, 2000. ISQED 2000. Proceedings. IEEE 2000 First International Symposium on*, 2000, pp. 173–179.
- [31] P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, "A test vector inhibiting technique for low energy bist design," in *VLSI Test Symposium, 1999. Proceedings. 17th IEEE*, 1999, pp. 407–412.
- [32] D. Gizopoulos, A. Pachalis, Y. Zorian, and M. Psarakis, "An effective bist scheme for arithmetic logic units," in *Test Conference, 1997. Proceedings., International*, 1997, pp. 868–877.
- [33] D. Gizopoulos, A. Paschalis, and Y. Zorian, "An effective built-in self-test scheme for parallel multipliers," *Computers, IEEE Transactions on*, vol. 48, no. 9, pp. 936–950, 1999.
- [34] L. Goldstein, "Controllability/observability analysis of digital circuits," *Circuits and Systems, IEEE Transactions on*, vol. 26, no. 9, pp. 685–693, 1979.
- [35] P. T. Gonciari, B. M. Al-Hashimi, and N. Nicolici, "Variable-length input huffman coding for system-on-a-chip test," *IEEE Trans. Comput.-Aided Des.*, vol. 22, no. 6, pp. 783–796, Jun. 2003.
- [36] I. Hamzaoglu and J. Patel, "Reducing test application time for full scan embedded cores," in *Fault-Tolerant Computing, 1999. Digest of Papers. Twenty-Ninth Annual International Symposium on*, 1999, pp. 260–267.
- [37] H. Hashempour, F. Meyer, F. Lombardi, and F. Karimi, "Hybrid multisite testing at manufacturing," in *Test Conference, 2003. Proceedings. ITC 2003. International*, vol. 1, 2003, pp. 927–936.

- [38] S. Hellebrand, H.-G. Liang, and H. Wunderlich, “A mixed mode bist scheme based on reseeding of folding counters,” in *Test Conference, 2000. Proceedings. International*, 2000, pp. 778–784.
- [39] S. Hellebrand, J. Rajski, S. Tarnick, S. Venkataraman, and B. Courtois, “Built-in test for circuits with scan based on reseeding of multiple-polynomial linear feedback shift registers,” *Computers, IEEE Transactions on*, vol. 44, no. 2, pp. 223–233, 1995.
- [40] G. Hetherington, T. Fryars, N. Tamarapalli, M. Kassab, A. Hassan, and J. Rajski, “Logic bist for large industrial designs: real issues and case studies,” in *Test Conference, 1999. Proceedings. International*, 1999, pp. 358–367.
- [41] D. Huffman, “Run-length encoding,” *IEEE Trans. Info Theory*, no. IT-12, pp. 399–401, Jul.
- [42] ———, “A method for the construction of minimum-redundancy codes,” *Proc. of the IRE*, vol. 40, no. 9, pp. 1098–1101, Sept. 1952.
- [43] B. Ireland and J. Marshall, “Matrix method to determine shift-register connections for delayed pseudorandom binary sequences,” *Electronics Letters*, vol. 4, no. 21, pp. 467–468, 1968.
- [44] J. T. Janusz Rajski, *Arithmetic built-in self-test for embedded systems*. Prentice-Hall, Englewood Cliffs, NJ, 1998.
- [45] A. Jas, J. Ghosh-Dastidar, M.-E. Ng, and N. A. Touba, “An efficient test vector compression scheme using selective huffman coding,” *IEEE Trans. Comput.-Aided Des.*, vol. 22, no. 6, pp. 797–806, Jun. 2003.
- [46] A. Jas, J. Ghosh-Dastidar, and N. A. Touba, “Scan vector compression/decompression using statistical coding,” in *Proc. VTS*, 1999, pp. 114–120.
- [47] A. Jas and N. Touba, “Test vector decompression via cyclical scan chains and its application to testing core-based designs,” in *Test Conference, 1998. Proceedings., International*, 1998, pp. 458–464.
- [48] S. Kajihara, K. Ishida, and K. Miyase, “Test vector modification for power reduction during scan testing,” in *VLSI Test Symposium, 2002. (VTS 2002). Proceedings 20th IEEE*, 2002, pp. 160–165.
- [49] E. Kalligeros, D. Kaseridis, X. Kavousianos, and D. Nikolos, “Reseeding-based test set embedding with reduced test sequences,” in *Quality of Electronic Design, 2005. ISQED 2005. Sixth International Symposium on*, 2005, pp. 226–231.
- [50] E. Kalligeros, X. Kavousianos, and D. Nikolos, “Multiphase bist: a new reseeding technique for high test-data compression,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 23, no. 10, pp. 1429–1446, 2004.

- [51] —, “Efficient multiphase test set embedding for scan-based testing,” in *Quality Electronic Design, 2006. ISQED '06. 7th International Symposium on*, 2006, pp. 432–438.
- [52] R. Kapur, R. Chandramouli, and T. W. Williams, “Strategies for low-cost test,” *IEEE Des. Test*, vol. 18, no. 6, pp. 47–54, Nov. 2001. [Online]. Available: <http://dx.doi.org/10.1109/54.970423>
- [53] X. Kavousianos, E. Kalligeros, and D. Nikolos, “Multilevel huffman coding: An efficient test-data compression method for ip cores,” *IEEE Trans. Comput.-Aided Des.*, vol. 26, no. 6, pp. 1070–1083, Jun. 2007.
- [54] —, “Optimal selective huffman coding for test-data compression,” *IEEE Trans. on Comput.*, vol. 56, no. 8, pp. 1146–1152, Aug. 2007.
- [55] —, “Multilevel-huffman test-data compression for ip cores with multiple scan chains,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 16, no. 7, pp. 926–931, 2008.
- [56] —, “Test data compression based on variable-to-variable huffman encoding with codeword reusability,” *IEEE Trans. Comput.-Aided Des.*, vol. 27, no. 7, pp. 1333–1338, 2008.
- [57] H. Ko and N. Nicolici, “Automated scan chain division for reducing shift and capture power during broadside at-speed test,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 27, no. 11, pp. 2092–2097, 2008.
- [58] B. Koenemann, “Lfsr-coded test patterns for scan designs,” in *Proc. ETS/ETC, VDE Verlag*, 1991, pp. 237–242.
- [59] B. Koenemann, C. Barnhart, B. Keller, T. Snethen, O. Farnsworth, and D. Wheeler, “A smartbist variant with guaranteed encoding,” in *Test Symposium, 2001. Proceedings. 10th Asian*, 2001, pp. 325–330.
- [60] C. V. Krishna, A. Jas, and N. A. Touba, “Test vector encoding using partial lfsr reseeding,” in *Proc. ITC*, 2001, pp. 885–893.
- [61] —, “Achieving high encoding efficiency with partial dynamic lfsr reseeding,” *ACM Trans. Des. Autom. of Electr. Syst.*, vol. 9, no. 4, pp. 500–516, Oct. 2004.
- [62] C. V. Krishna and N. Touba, “Reducing test data volume using lfsr reseeding with seed compression,” in *Test Conference, 2002. Proceedings. International*, 2002, pp. 321–330.
- [63] —, “Adjustable width linear combinational scan vector decompression,” in *Computer Aided Design, 2003. ICCAD-2003. International Conference on*, 2003, pp. 863–866.

- [64] J. Lee and N. A. Touba, "Lfsr-reseeding scheme achieving low-power dissipation during test," *IEEE Trans. Comput.-Aided Des.*, vol. 26, no. 2, pp. 396–401, Feb. 2007.
- [65] J. Lee and N. Touba, "Low power test data compression based on lfsr reseeding," in *Computer Design: VLSI in Computers and Processors, 2004. ICCD 2004. Proceedings. IEEE International Conference on*, 2004, pp. 180–185.
- [66] K.-J. Lee, J.-J. Chen, and C.-H. Huang, "Using a single input to support multiple scan chains," in *Computer-Aided Design, 1998. ICCAD 98. Digest of Technical Papers. 1998 IEEE/ACM International Conference on*, 1998, pp. 74–78.
- [67] L.-J. Lee, W.-D. Tseng, R.-B. Lin, and C.-H. Chang, " $2^n$  pattern run-length for test data compression," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 31, no. 4, pp. 644–648, 2012.
- [68] J. Li, Q. Xu, Y. Hu, and X. Li, "ifill: An impact-oriented x-filling method for shift-and capture-power reduction in at-speed scan-based testing," in *Design, Automation and Test in Europe, 2008. DATE '08*, 2008, pp. 1184–1189.
- [69] L. Li and K. Chakrabarty, "Test data compression using dictionaries with fixed-length indices [soc testing]," in *VLSI Test Symposium, 2003. Proceedings. 21st*, 2003, pp. 219–224.
- [70] L. Li, K. Chakrabarty, S. Kajihara, and S. Swaminathan, "Efficient space/time compression to reduce test data volume and testing time for ip cores," in *Proc. ICVD*, 2005, pp. 53–58.
- [71] W. Li, S. Reddy, and I. Pomeranz, "On reducing peak current and power during test," in *VLSI, 2005. Proceedings. IEEE Computer Society Annual Symposium on*, 2005, pp. 156–161.
- [72] S.-P. Lin, C.-L. Lee, J.-E. Chen, J.-J. Chen, K.-L. Luo, and W.-C. Wu, "A multilayer data copy test data compression scheme for reducing shifting-in power for multiple scan design," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 15, no. 7, pp. 767–776, 2007.
- [73] J. Marshall, B. Ireland, B. Bajoga, and K. Latawiec, "New method of generation of shifted linear pseudorandom binary sequences," *Electrical Engineers, Proceedings of the Institution of*, vol. 122, no. 4, pp. 448–, 1975.
- [74] E. J. McCluskey, *Logic design principles - with emphasis on testable semicustom circuits*, ser. Prentice Hall series in computer engineering. Prentice Hall, 1986.
- [75] S. Mitra and K. S. Kim, "Xpand: an efficient test stimulus compression technique," *Computers, IEEE Transactions on*, vol. 55, no. 2, pp. 163–173, 2006.

- [76] K. Miyase, S. Kajihara, and S. Reddy, "Multiple scan tree design with test vector modification," in *Test Symposium, 2004. 13th Asian*, 2004, pp. 76–81.
- [77] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, April 1965.
- [78] G. Mrugalski, J. Rajski, D. Czysz, and J. Tyszer, "New test data decompressor for low power applications," in *Proc. DAC*, 2007, pp. 539–544.
- [79] G. Mrugalski, J. Rajski, and J. Tyszer, "Ring generators - new devices for embedded test applications," *IEEE Trans. Comput.-Aided Des.*, vol. 23, no. 9, pp. 1306–1320, Sept. 2004.
- [80] G. Mrugalski, J. Tyszer, and J. Rajski, "Linear independence as evaluation criterion for two-dimensional test pattern generators," in *VLSI Test Symposium, 2000. Proceedings. 18th IEEE*, 2000, pp. 377–386.
- [81] N. Nicolici and B. Al-Hashimi, *Power-Constrained Testing of VLSI Circuits*. Kluwer Academic, Norwell, MA, 2003.
- [82] M. Nourani and M. H. Tehranipour, "Rl-huffman encoding for test compression and power reduction in scan applications," *ACM Trans. Des. Autom. of Electr. Syst.*, vol. 10, pp. 91–115, Jan. 2005.
- [83] A. Pandey and J. Patel, "Reconfiguration technique for reducing test time and test data volume in illinois scan architecture based designs," in *VLSI Test Symposium, 2002. (VTS 2002). Proceedings 20th IEEE*, 2002, pp. 9–15.
- [84] W. H. M. Paul H. Bardell, "Self-testing of multichip logic modules," in *ITC*, 1982, pp. 200–204.
- [85] B. Pouya and A. Crouch, "Optimization trade-offs for vector volume and test power," in *Test Conference, 2000. Proceedings. International*, 2000, pp. 873–881.
- [86] J. Rajski, N. Tamarapalli, and J. Tyszer, "Automated synthesis of phase shifters for built-in self-test applications," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 19, no. 10, pp. 1175–1188, 2000.
- [87] J. Rajski and J. Tyszer, "Design of phase shifters for bist applications," in *VLSI Test Symposium, 1998. Proceedings. 16th IEEE*, 1998, pp. 218–224.
- [88] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, "Embedded deterministic test," *IEEE Trans. Comput.-Aided Des.*, vol. 23, no. 5, pp. 776–792, 2004.
- [89] W. Rao, I. Bayraktaroglu, and A. Orailoglu, "Test application time and volume compression through seed overlapping," in *Design Automation Conference, 2003. Proceedings*, 2003, pp. 732–737.

- [90] S. Reda and A. Orailoglu, "Reducing test application time through test data mutation encoding," in *Proc. DATE*, 2002, pp. 387–393.
- [91] S. Reddy, K. Miyase, S. Kajihara, and I. Pomeranz, "On test data volume reduction for multiple scan chain designs," in *VLSI Test Symposium, 2002. (VTS 2002). Proceedings 20th IEEE*, 2002, pp. 103–108.
- [92] S. Remersaro, X. Lin, S. Reddy, I. Pomeranz, and J. Rajski, "Scan-based tests with low switching activity," *Design Test of Computers, IEEE*, vol. 24, no. 3, pp. 268–275, 2007.
- [93] S. Remersaro, X. Lin, Z. Zhang, S. Reddy, I. Pomeranz, and J. Rajski, "Preferred fill: A scalable method to reduce capture power for scan based designs," in *Test Conference, 2006. ITC '06. IEEE International*, 2006, pp. 1–10.
- [94] P. Rosinger, P. Gonciari, B. Al-Hashimi, and N. Nicolici, "Simultaneous reduction in volume of test data and power dissipation for systems-on-a-chip," *Electronics Letters*, vol. 37, no. 24, pp. 1434–1436, 2001.
- [95] S. Samaranayake, E. Gizdarski, N. Sitchinava, F. Neuveux, R. Kapur, and T. Williams, "A reconfigurable shared scan-in architecture," in *VLSI Test Symposium, 2003. Proceedings. 21st*, 2003, pp. 9–14.
- [96] K. Sankaralingam, R. Oruganti, and N. Touba, "Static compaction techniques to control scan vector power dissipation," in *VLSI Test Symposium, 2000. Proceedings. 18th IEEE*, 2000, pp. 35–40.
- [97] J. Saxena, K. M. Butler, V. B. Jayaram, S. Kundu, N. V. Arvind, P. Sreeprakash, and M. Hachinger, "A case study of ir-drop in structured at-speed testing," in *Proc. ITC*, vol. 1, Oct. 2003, pp. 1098–1104.
- [98] L. Schafer, R. Dorsch, and H. Wunderlich, "Respin++ - deterministic embedded test," in *Test Workshop, 2002. Proceedings. The Seventh IEEE European*, 2002, pp. 37–44.
- [99] S.-W. Seong and P. Mishra, "Bitmask-based code compression for embedded systems," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 27, no. 4, pp. 673–685, 2008.
- [100] M. Shah and J. Patel, "Enhancement of the illinois scan architecture for use with multiple scan inputs," in *VLSI, 2004. Proceedings. IEEE Computer society Annual Symposium on*, 2004, pp. 167–172.
- [101] C. Shi and R. Kapur, "How power-aware test improves reliability and yield," *EE Times EDA news online*, 09/15/2004.

- [102] N. Sitchinava, E. Gizdarski, S. Samaranayake, F. Neuveux, R. Kapur, and T. Williams, “Changing the scan enable during shift,” in *VLSI Test Symposium, 2004. Proceedings. 22nd IEEE*, 2004, pp. 73–78.
- [103] C. Stroud, *A Designer’s Guide to Built-In Self-Test*. Springer, Boston, MA, 2002.
- [104] ———, “An automated bist approach for general sequential logic synthesis,” in *Design Automation Conference, 1988. Proceedings., 25th ACM/IEEE*, 1988, pp. 3–8.
- [105] H. Tang, S. Reddy, and I. Pomeranz, “On reducing test data volume and test application time for multiple scan chain designs,” in *Test Conference, 2003. Proceedings. ITC 2003. International*, vol. 1, 2003, pp. 1079–1088.
- [106] M. H. Tehranipour, M. Nourani, and K. Chakrabarty, “Nine-coded compression technique for testing embedded cores in socs,” *IEEE Trans. Very Large Scale Integr.*, vol. 13, no. 6, pp. 719–731, Jun. 2005.
- [107] M. Tehranipour, M. Nourani, K. Arabi, and A. Afzali-Kusha, “Mixed rl-huffman encoding for power reduction and data compression in scan test,” in *Circuits and Systems, 2004. ISCAS ’04. Proceedings of the 2004 International Symposium on*, vol. 2, 2004, pp. II-681–4 Vol.2.
- [108] V. Tenentes, X. Kavousianos, and E. Kalligeros, “State skip lfsrs: Bridging the gap between test data compression and test set embedding for ip cores,” in *Design, Automation and Test in Europe, 2008. DATE ’08*, 2008, pp. 474–479.
- [109] ———, “Single and variable-state-skip lfsrs: Bridging the gap between test data compression and test set embedding for ip cores,” *IEEE Trans. Comput.-Aided Des.*, vol. 29, no. 10, pp. 1640–1644, Oct. 2010.
- [110] N. A. Touba, “Survey of test vector compression techniques,” *IEEE Design & Test*, vol. 23, no. 4, pp. 294–303, Apr. 2006.
- [111] N. Touba, “Circular bist with state skipping,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 10, no. 5, pp. 668–672, 2002.
- [112] J. Tyszer, D. Czysz, G. Mrugalski, N. Mukherjee, and J. Rajski, “On deploying scan chains for data storage in test compression environment,” *IEEE Design & Test*, Early access article, 2012.
- [113] H. Vergos, D. Nikolos, M. Bellos, and C. Efstathiou, “Deterministic bist for rns adders,” *Computers, IEEE Transactions on*, vol. 52, no. 7, pp. 896–906, 2003.
- [114] E. Volkerink, A. Khoche, L. Kamas, J. Rivoir, and H. Kerkhoff, “Tackling test trade-offs from design, manufacturing to market using economic modeling,” in *Test Conference, 2001. Proceedings. International*, 2001, pp. 1098–1107.

- [115] E. Volkerink, A. Khoche, and S. Mitra, "Packet-based input test data compression techniques," in *Test Conference, 2002. Proceedings. International*, 2002, pp. 154–163.
- [116] E. Volkerink, A. Khoche, J. Rivoir, and K.-D. Hilliges, "Test economics for multi-site test with modern cost reduction techniques," in *VLSI Test Symposium, 2002. (VTS 2002). Proceedings 20th IEEE*, 2002, pp. 411–416.
- [117] E. Volkerink and S. Mitra, "Efficient seed utilization for reseeding based compression," in *VLSI Test Symposium, 2003. Proceedings. 21st*, 2003, pp. 232–237.
- [118] H. Vranken, F. Hapke, S. Rogge, D. Chindamo, and E. Volkerink, "Atpg padding and ate vector repeat per port for reducing test data volume," in *Proc. ITC*, 2003, pp. 1069–1078.
- [119] L.-T. Wang, X. Wen, H. Furukawa, F.-S. Hsu, S.-H. Lin, S.-W. Tsai, K. Abdel-Hafez, and S. Wu, "Virtualscan: a new compressed scan technology for test cost reduction," in *Test Conference, 2004. Proceedings. ITC 2004. International*, 2004, pp. 916–925.
- [120] L.-T. Wang, K. Abdel-Hafez, X. Wen, B. Sheu, S. Wu, S.-H. Lin, and M.-T. Chang, "Ultrascan: using time-division demultiplexing/multiplexing (tddm/tdm) with virtualscan for test cost reduction," in *Test Conference, 2005. Proceedings. ITC 2005. IEEE International*, 2005, pp. 946–953.
- [121] L.-T. Wang, C.-W. Wu, and X. Wen, *VLSI Test Principles and Architectures: Design for Testability (Systems on Silicon)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2006.
- [122] S. Wang and S. K. Gupta, "Ds-lfsr: A new bist tpg for low heat dissipation," in *Proceedings of the 1997 IEEE International Test Conference*, ser. ITC '97. Washington, DC, USA: IEEE Computer Society, 1997, pp. 848–. [Online]. Available: <http://dl.acm.org/citation.cfm?id=844384.845758>
- [123] Z. Wang and K. Chakrabarty, "Test data compression using selective encoding of scan slices," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 16, no. 11, pp. 1429–1440, 2008.
- [124] S. Ward, C. Schattauer, and N. Touba, "Using statistical transformations to improve compression for linear decompressors," in *Defect and Fault Tolerance in VLSI Systems, 2005. DFT 2005. 20th IEEE International Symposium on*, 2005, pp. 42–50.
- [125] X. Wen, K. Miyase, S. Kajihara, H. Furukawa, Y. Yamato, A. Takashima, K. Noda, H. Ito, K. Hatayama, T. Aikyo, and K. Saluja, "A capture-safe test generation scheme for at-speed scan testing," in *Test Symposium, 2008 13th European*, 2008, pp. 55–60.

- [126] X. Wen, S. Kajihara, K. Miyase, T. Suzuki, K. Saluja, L.-T. Wang, K. Abdel-Hafez, and K. Kinoshita, "A new atpg method for efficient capture power reduction during scan testing," in *VLSI Test Symposium, 2006. Proceedings. 24th IEEE*, 2006, pp. 6 pp.–65.
- [127] X. Wen, K. Miyase, T. Suzuki, S. Kajihara, L.-T. Wang, K. Saluja, and K. Kinoshita, "Low capture switching activity test generation for reducing ir-drop in at-speed scan testing," *J. Electron. Test.*, vol. 24, no. 4, pp. 379–391, Aug 2008.
- [128] X. Wen, K. Miyase, T. Suzuki, Y. Yamato, S. Kajihara, L.-T. Wang, and K. Saluja, "A highly-guided x-filling method for effective low-capture-power scan test generation," in *Computer Design, 2006. ICCD 2006. International Conference on*, 2006, pp. 251–258.
- [129] X. Wen, Y. Yamashita, S. Morishima, S. Kajihara, L.-T. Wang, K. Saluja, and K. Kinoshita, "Low-capture-power test generation for scan-based at-speed testing," in *Test Conference, 2005. Proceedings. ITC 2005. IEEE International*, 2005, pp. 10 pp.–1028.
- [130] T. W. Williams and N. C. Brown, "Defect level as a function of fault coverage," *IEEE Trans. Comput.*, vol. 30, no. 12, pp. 987–988, Dec. 1981. [Online]. Available: <http://dx.doi.org/10.1109/TC.1981.1675742>
- [131] P. Wohl, J. Waicukauski, S. Patel, F. DaSilva, T. Williams, and R. Kapur, "Efficient compression of deterministic patterns into multiple prpg seeds," in *Test Conference, 2005. Proceedings. ITC 2005. IEEE International*, 2005, pp. 10 pp.–925.
- [132] G. Wolff and C. Papachristou, "Multiscan-based test compression and hardware decompression using lz77," in *Proc. ITC*, 2002, pp. 331–339.
- [133] A. Wurtenberger, C. Tautermann, and S. Hellebrand, "Data compression for multiple scan chains using dictionaries with corrections," in *Test Conference, 2004. Proceedings. ITC 2004. International*, 2004, pp. 926–935.
- [134] M. Yi, H. Liang, L. Zhang, and W. Zhan, "A novel x-ploiting strategy for improving performance of test data compression," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 18, no. 2, pp. 324–329, 2010.
- [135] N. Zacharia, J. Rajski, and J. Tyszer, "Decompression of test data using variable-length seed lfsrs," in *VLSI Test Symposium, 1995. Proceedings., 13th IEEE*, 1995, pp. 426–433.
- [136] G. Zeng and H. Ito, "Concurrent core test for soc using shared test set and scan chain disable," in *Proc. DATE*, 2006, pp. 1–6.

- [137] Q. Zhou and K. Balakrishnan, "Test cost reduction for soc using a combined approach to test data compression and test scheduling," in *Proc. DATE*, 2007, pp. 1–6.
- [138] Y. Zorian, "Testing the monster chip," *Spectrum, IEEE*, vol. 36, no. 7, pp. 54–60, 1999.