Deep Science Publishing
https://doi.org/10.70593/978-81-989050-5-5_10

DeepScience
Open Access Books

Chapter 10: High-precision validation and multi-domain testing techniques for complex chipsets

10.1. Introduction

With the proliferation of digital devices used in the consumer space, there has been an increased demand for performance and a concurrent growth in power and cost that needs to be effectively managed for sustaining the growth in the semiconductor industry. Chip manufacturers are continuously challenged with the need to reduce the risk of test escapes as they begin the migration at both the die level as well as the package level by incorporating unproven and unverified technology in production systems (Goldsmith, 2005; Biglieri, 2007; Heath et al., 2016). This is also being emphasized by the technology transition coming from 20 nm to 14nm and further migrations to 10nm and below where chip costs are extremely high and the silicon brought up is concurrent with product implementation timelines that necessitate the introduction of effective validation and testing strategies that are organized and method driven. As has been experienced in the past, as technology scales and design complexities increase, multi-domain functional testing of die with deeply embedded die-to-die and die-to-package interconnect circuitry becomes critical in ensuring adequate test coverage and reducing the risk of test escapes from manufacturing. Emphasis on the need for the early introduction of Validation and Fault Testing techniques is becoming increasingly important. We present techniques and concepts that help facilitate the reduced validation time frame and validate complex die with high density die-to-die and die-to-package interconnects and circuitry. These techniques are modular in implementation and can deal with complex test access challenges during silicon bring up and preliminary validation when standard boundary scan and serial link circuitry techniques cannot be reliably used (Tse & Viswanath, 2005; Larsson et al., 2014; Heath et al., 2016).

10.2. Overview of Chipset Architecture

A modern computer system is made up of one or more processors, a set of memory manager, caches, control units, and Input-output (I/O) peripherals such as an interface to the disk, a display device, network interfaces, and so on. To connect all these functional units together, external buses are used. The number of cycles that the processor needs to complete a function, such as issuing a read command for a certain memory address, may be small, and the number of general-purpose processors that perform this function is a very small number. This is the motivation of building a small number of processors with a large number of external buses and using a set of interfaces for all the I/O needs of the system. These interfaces are connected to the chipset through parallel buses which translate the parallel buses to the appropriate format used at the external buses of the CPU. The chipset then connects to the graphic rendering unit, memory units, storage, networking, and other peripherals.

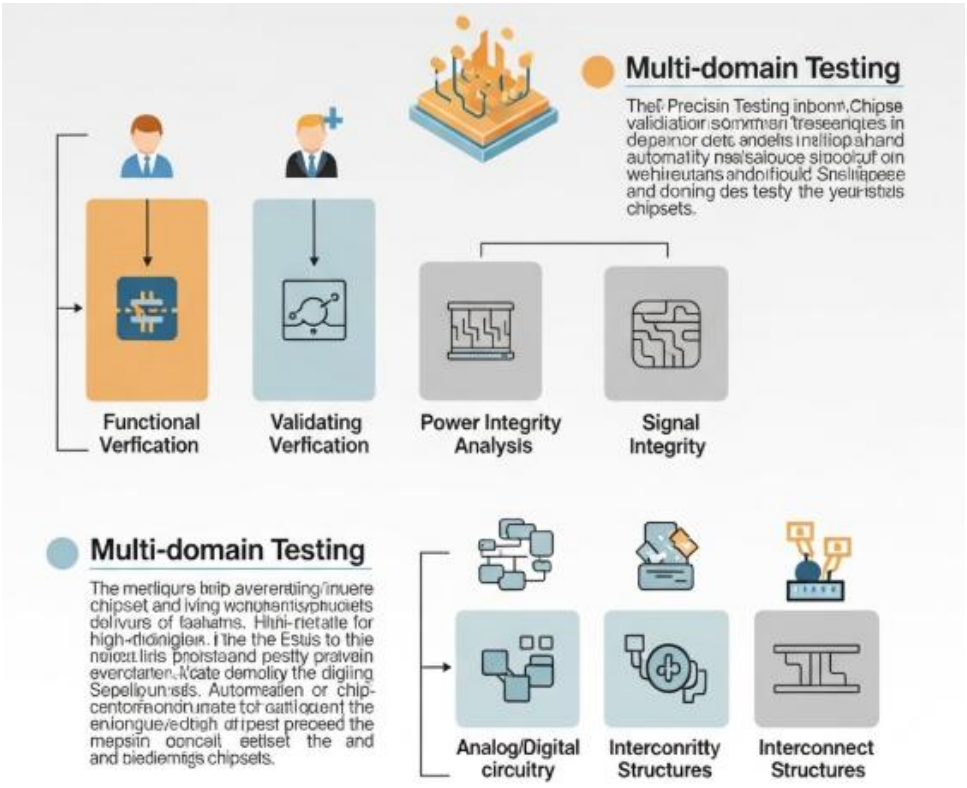

Fig 10.1: High-Precision Validation and Multi-Domain Testing

To provide high speeds of operation, the integrated chipset provides built-in high-speed communication links among the processing units connected in the system. The core components of these integrated chipsets are often several small-scale integrated circuits connected into a Chip Set. These core components of the chipset are made of various

digital processing bricks such as bus controllers, programmable logic devices, memory controllers, network interface controllers, bus interfaces, address decoders and I/O port controllers working in an integrated manner. Compared to a processor, these peripherals are programmable, and implement many more functions, but work simultaneously for a very small fraction of the system's actual operational time. Hence, the cost of these I/O processor circuits is less than the processor circuits in an equivalent area; and so, many more logic gates are available in the same amount of area, power, and time.

10.3. Importance of Validation in Chipset Development

Chipset development is a fast-paced and intense process that typically lasts only a few years. This is not typically long enough to allow for discovery of corner-case bugs by normal product user experience. Realistically, the product is not deployed in any truly long-term, stressing environments, and thus those important corner-case bugs are usually found only many years into the product life cycle - and often during the development of the next generation chipset. No-blame validation is extremely helpful for this problem. No-blame validation is designed to detect those important corner case bugs early on by focusing on creating relatively simple, relatively short tests that take an "end-user perspective" of functionality. No blame validation thus has a strong emphasis on Task Execution Completion, SEC - Self Error Checking, and YET - User-Experience Time targets. It is foolish to focus validation on million-operation once-in-a-while stress tests - those are definitely designed to be bug initiators but discovering a bug by executing a difficult special test months or years after it was feared to be dormant is not a desired objective.

The biggest chip/device buggers are the ones that are the most unexpected and "found by accident"; these types of bugs cause the most pain and are usually the hardest to debug and recreate. Strong peripheral interaction exploratory testing methods help catch those types of bugs early. Chipset devices are an autonomous system containing many devices, including primary processors, secondary processors, sensors, actuators, wireless devices, and GUI and I/O hardware. These devices can last through much product development but not as long as the product itself.

10.4. High-Precision Validation Techniques

To accomplish multi-domain validation of a chipset, it is mandatory to validate all functionalities exposed by the chipset to all external agents as correctly specified and to fulfill the required specifications within the allowed limits. For example, Functional Validation establishes that all input/output signal logics are applied correctly and that correct logics are sensed from the outputs. Timing Validation consists in assessing that

timing for a signal appearing at the output is correct and inspected with respect to the timing of the respective input signal being inspected. Moreover, Timing Validation must be performed under all allowed operation scenarios. Similarly, Power Validation consists in verifying that the power consumed under all possible functional cases does not exceed the specified upper limit.

The set of methodologies used to verify all of the possible device operating cases with the target precision is termed as High-Precision Validation Techniques. High-Precision Validation Techniques may be classified into (1) static and (2) dynamic validation techniques. Static Techniques do not require any operation to assess the validity of the device design within the required limits. Rather than this, they utilize model analysis on all functions related to validation to demonstrate that no error can occur. Dynamic Techniques involve actual operation of the device in order to observe the power, timing relationships, and Functional statistics.

Fig 10.3: High-Precision Validation Techniques

Static validation techniques enable the highest validation precision possible because there is no uncertainty related to the test applied, detection of the expected output, and

sensitivity to outer perturbations. For timing validation, this approach assures the maximum accuracy in timing measurement because it does not suffer from any restriction on the edges of the signals used for timing measurements. Static techniques mainly include (1) simulation-based bitmap analysis, (2) SAT-based verification, (3) Symbolic Transfer-Based Method, (4) timing clock gating, (5) timing cone of influence analysis, (6) symbolic functional analysis, and (7) static power analysis.

10.4.1. Static Validation Methods

The need for ever more complex DSP/FPGA-based chipsets in various domains is rising tremendously. Moreover, both hardware componentes, as well as associated and embedded software for functional validation go complex, huge, and excessive, making it really challenging, tedious, and by far mostly impossible for design houses to ensure without any doubt that their products are designed correctly, that there are no hidden silicon implementation flaws, no bugs in the IPs, hardware and software works perfect as a single product valid for the intended application domain, chips-based chipset address and serve the application correctly.

In parallel, not being able to provide technical, valid, well-advanced validation and test techniques and tools represents a distinctive limit for third parties companies who are investing time and efforts in licensing and utilizing costly blocks soldered in complex assemblies. Though sometimes anyhow locked to a specific third-party house. On one side, traditional validation methods are basically returning zero debug capabilities for such complex heterogeneous designs, on the other side, resorts to ad hoc internal developed or small-house-developed tools cannot address high-performance verification.

The only valid resort on these problems is to capitalize on existing static validation previously developed, continuously updated, and real-life-proven methods and associated software tools that take into account the typical core-design validation problems of huge complex heterogeneous DSP/FPGA-based circuitry architectures at multi-DSP, distributed processing level. In particular, both advanced ASIC physical design closed-topology methods and validation and test design hierarchical methodology for DSP/FPGA—high-precision validation of architectural solutions are considered, that are exploiting both usage and tech at this assembly level.

10.4.2. Dynamic Validation Approaches

The definition of a functional validation metric is a necessary and sufficient condition to formally establish the precision of any dynamic validation method, under given

assumptions. As a first ad-hoc step towards high-precision validation methods, initial techniques focused on implementing expressive directed test generation techniques to reduce the gap between testing and functional validation. The proposed approaches restrict the search space for functional validation to specific scenarios where the functional validation metric density is maximally favorable to the method itself.

Directed state-space search adds a validation mode on top of the detection capability of a general DSP. By using combinations of known input data, the controller of the selected DSP can optimally stretch functional validation and trigger the error. By itself, functional validation is always a non-deterministic process; several VLSI deployment aspects, e.g. operating conditions and process variations, errors generated by other non-validated chips in the same package, etc., can collaborate in hiding the fault effect into noise, preventing detection. By intentionally steering the operation conditions of the chip, the probability density of the errors produced at its output data is altered. The chip detection is temporarily maximized, allowing detection of errors that would otherwise remain invisible due to non-determinism. Dynamic methods are implemented in most cases after final deployment in order to target and test undesired behaviors.

Since the 1980's, the concept of dynamic methods has been expanded towards post production implementation, initially taking physical form in the semi-invasive Active Load Testing. The idea was that one or more good chips are either purposely connected to the faulty chip terminal I/O ports or forced to a calibration state. By minimizing the difference between the measured and predicted signals, the influence of fabrication and design errors could be evaluated.

10.5. Multi-Domain Testing Strategies

As the silicon technology moved from planar to 3D FinFET structures, the complexity of integrated circuits increased dramatically. Today, the modern chips contain multiple design domains ranging from dozens of domains of many heterogeneous digital blocks to high-performance, mixed-signal or analog blocks. Naturally, the Test methods and design-for-test structures have to match this growing complexity. The classification is usually in digital or analog domains. Digital domain test deals with testing of conventional combinational or sequential logic using a standard extended deterministic test approach. The issues include controllability, observability, stuck-at faults for fast-path digital blocks, and access for sequential blocks. Test generation is simplified based on built-in test structures such as scan. The correct test or fault coverage is critical to failure prediction for complex designs. For fast-path digital blocks placed into the Mixed-Signal Domain, pattern generation is complicated as test input vectors need to drive analog signals such as AC or DC bias notes.

Analog domain test has been around for several decades, and comprises testing blocks such as A/D or D/A converters, PLLs, DACs, operational and passive filters, amplifiers, comparators, and buffers. The methods used are based on special characteristics or transfer functions tied to the block. Digital-based testers are generally used for hybrid-type DFT. More rigid test procedures are required that may include signal integrity analysis of the device under test with careful selection of AC or DC patterns injected at input test pins. Mixed-Signal Domain Test Strategies test partitions for hybrid chips where only some of the blocks are mixed-signal or analog. These strategies are based on simplistically regarding the mixed-signal blocks as being digital-testable. Such partitioning constraints have been used in system on chip designs based on the use of passive mixed-signal block DFT solutions.

10.5.1. Digital Domain Testing

Most multi-domain techniques explore the digital testing of the mixed-signal blocks. Due to this fact, we must first discuss the existing digital testing techniques, with a joint solution proposed in the age of the emerging complex multi-domain systems. Most digital blocks are tested by faults in the circuit.

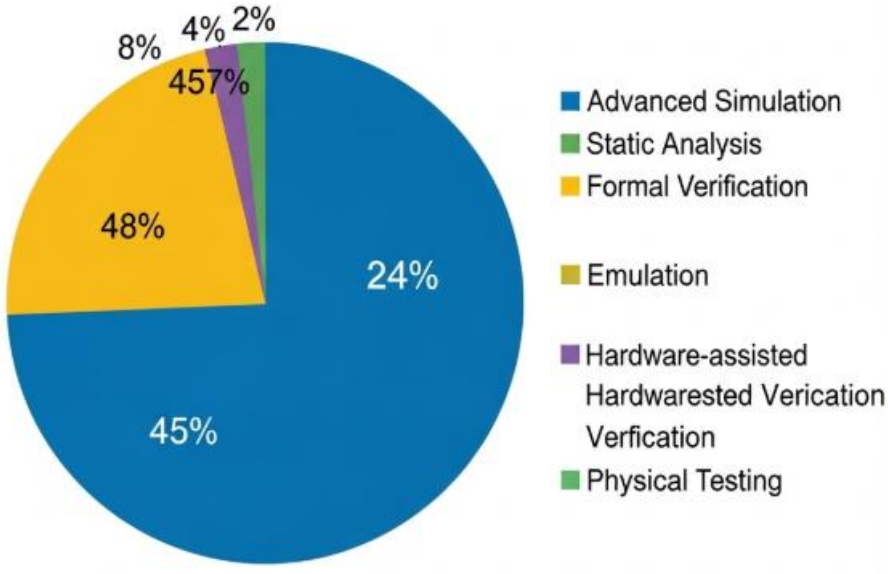

Fig 10.3: Multi-Domain Testing Techniques for Complex Chipsets

The fault models can be defined at different levels of abstraction, such as: the layout level, the netlist level, the transistor level. The most common practice is to define a fault list – either by simulation or learning from real failures. The fault coverage is then determined from the percentage of detected faults. If detection is not possible, design

changes or additional testing strategies are employed in order to raise the fault coverage. Most tests are created with the use of Automatic Test pattern Generation tools. A test is said to have structural fault coverage if it detects certain types of faults mostly associated with structural problems in logic circuits. Faults that are never detected by the test set are referred to as redundant faults. If in addition the test satisfies all the logic detection states for certain types of fault modeling, it is said to have logical fault coverage. Logic fault simulation has traditionally been the most widely used approach for test generation. Due to the structural simplicity of digital circuits and the ease of modifying the descriptions, there are tools that require only a short time to simulate all possible circuit input combinations. However, this technique is prohibitive in terms of computation time, for the case of large circuits. There are special tools designed to work on large circuits, but they are still quite slow.

10.5.2. Analog Domain Testing

The focus of this section is on fault modeling and ATPG for analog circuits, as well as their practical implementations. This testing is critical because in most chipsets there are a significant number of analog blocks such as A/D and D/A converters, PLLs, ADCs, DACs, CMOS amplifiers, RF circuits, etc. All these circuits are responsible for some important functions in a digital image processing system. If the performance of any of these analog circuits turns out to be below specification, the overall functionality of the chipset could be seriously affected. As analog testing is a major bottleneck during chip production, research work for developing efficient techniques for testing these important blocks of the system is currently a very important field of interest.

Fault modeling is a key area of research for both theoretical and practical aspects of analog circuit testing. In the past two decades, extensive research work has been done on the reliability and quality issues for analog and mixed-signal circuits. Some of the important aspects in this regard are semiconductor aging under high thermal environments, production problems in integrated circuits, electrostatic discharge problems, manufacturability and erosion of chip components, etc. In addition, many parameter variation problems are mostly prefeature problems, which are driven by design rules and circuit forced scaling. As a result, increasing system complexity makes it more and more difficult to conduct fault simulation and testing algorithms for all analog devices.

Most of the commercial tools available for analog fault modeling are based on multiport, small-signal models of linear and nonlinear circuits. There are several methods, which are based on perturbation of circuit parameters and generation of faulty frequency responses. In addition to this, several fault models have also been proposed for the non-linear devices based on physical structure or parameters like transistor-level behavioral

macromodels. These models are based on some physical parameters rather than simply voltage transfer characteristics.

10.5.3. Mixed-Signal Domain Testing

Within the context of multi-domain testing, this section covers strategies and techniques to test the Mixed-Signal domain. We shall elaborate on some of the existing methodologies as well as on a specific ADC-DAC concurrent testing strategy specifically conceived as a Mixed-Signal domain test. The Analog-Mixed Signal (AMS) test cost has become essential for the semiconductor industry, since AMS devices represent more than one-third of all semiconductor production – and the percentage is increasing. These devices are primarily the Analog-Digital Converters (ADC), Digital-Analog Converters (DAC), and Power Management Integrated Circuits (PMIC). To produce AMS Devices that surpass stringent quality and reliability standards with minimal economic impact, the industry has to develop efficient test methods for a wide variety of AMS. The extensive testing of AMS devices, mostly carried out at wafer test and final package test, including Long Test Programs on ATE, is driven by the Consumer Electronics sector. The demand for low-cost solutions with constant availability, is challenging AMS design-in test.

With the goal of reducing AMS test costs, the development of innovative concurrent Moving Device Under Test (DUT) architectures and design-in test strategies and methodologies, is ever more considered. To achieve this goal, the state of the art considers the design of dedicated AMS design architectures that could enable Device Under Test (DUT) Fault Modeling and how these reduce the need for extensive Independent Built-In Self-Test (IBIST) functions for each Analog & Mixed-Signal block, including Power Distribution Network (PDN) built-in self-test strategies for PMIC devices. The device-in test methods include concurrent Mixed-Signal Chain Test (MSCT) architectures and methods for Data Converters and Power Management Integrated Circuits. These methods present the capability to reduce the test cost significantly.

10.6. Test Automation Frameworks

A testing framework can be defined as a solution that provides a specific environment for developing, running, and managing test cases. High technology chip designs involve millions of logic elements, leading to a diverse set of features and capabilities. Testing the functionality of such large-scale integration designs requires the execution of an equally large number of tests. It is therefore essential to automate the hardware validation process to a large extent. Most products are equipped with internal capabilities that allow

the designer to assemble, configure and send data to the hardware in a streamlined manner. Other products also allow following the response in real-time at the speed of the hardware product. Integrated tools and flexible drivers transfer addresses, set parameters to the chip, read it back and check the answer. These let designers focus on writing tests using high-level languages rather than spending months developing it from low-level functions.

The production of hardware validation patterns are generated from the design decompilation. However, normally the designer has to go through an iterative process with the production team, refining the tests and driving the pattern generation tools multiple times until the generated patterns achieve satisfactory results. Some provide an assistance page which allows running in automated mode and takes full advantage of test frameworks capabilities. Such integration level of automation not only speeds up the hardware validation process but also eliminates the repetitive front-end design checks once brought out by the use of test software. Often these bugs or oversights could be difficult to show by automated functional checks. The test framework will visualize any discrepancy in response in a report format, showing input vectors and expected/actual chip responses.

10.6.1. Automated Test Development

Test automation these days has become essential for complex SoCs as it not only helps in completing the record number of test cases with shorter time but also helps in achieving the required functional and quality coverage. Creating test software for IPs and complete design validation requires a lot of effort on specific tooling but the requirement is justified as the design complexity increases. There are a lot of tools available today which offer various capabilities in the test development domain. The kind of support available on various aspects in test automation frameworks play a crucial role in deciding the framework which is chosen for the specific collaboration. The aspects include the support given for Hardware-Software Collaboration, System Modeling & Stimulus Generation, Forensic capabilities during Failure Analysis, Post Analysis, Debugger Support, Factory Enablement, and Multi Domain Support. The collaboration of Hardware Design, Verification and Test experts helps in achieving these capabilities. Test Development Tools today focus on multi-adoption capabilities for the ecosystem leading to less maintenance and faster time-to-market. Integrated Capabilities in the area of Automated Test Development along with Coverage Checking, Debugging and Revision Control plays a crucial role in cutting down the overall development time and optimizing Test Times.

The statement stands true for different aspects and a single internal organization is not capable of supporting everything. In-House Tools work well for supporting particular

areas related to Test Development In-House while Commercial Tools have their own pros and cons and it is often necessary to evaluate the risk and work on integrating both the capabilities in a single collaboration. The integration and collaboration in these areas cannot be understated as modern test development focuses on the Complete Lifecycle of Debugging, Scalable and Integrated Data Management and Fail Forensics with General-Purpose Tests with Lot-Level Screening, Composite Test Conditions, and High-Level Test Instantiation Interface with Centralized Scalable Data Management for Efficient and Flexible Access to Test Metadata for Test Planning, Retrieval, Planning windows, and Sign-Off while Unity of Structure and Methodology is Ensured by Comprehensive Guidelines for Test Metadata Structure.

10.6.2. Continuous Integration for Validation

Show me a validation team that doesn't want a Good-Enough Release; that team is probably oblivious to the goal of the Project Manager, or doesn't care about the impending product release! Validation is probably the only project phase that keeps moving right until after the product starts to sell. You are saving some of the worst of the worst bugs for the final release – bugs that are found by customers or by the field. Is there a better way? Continuous Integration helps eliminate bugs during the integration process. In typical software Continuous Integration, every night, the latest code from the repository is compiled, linked, and all unit tests are run. Results from the builds and tests are posted. Developers work during normal hours, and bugs are found and fixed before their intent disappears. In more integrated development processes, dev code is integrated twice a week, then run through additional tests. Continuous Integration applies these ideas to validation.

For hardware validation, Continuous Integration is at least partly scriptable (i.e. you set up a script to start a new build after a certain time, regardless of whether any changes made in the previous iteration). However, this is a monolithic idea. The common case of the last phase of silicon development that gently segues into product release doesn't combine well with "commit early, commit often". For hardware validation, Continuous Integration isn't. It doesn't have to do everything better than people manually doing it, just do the common case better. Have any steps that are manual, do those steps. Then checkpoint the manual phase, and scrutinize the delta between checkpoints. Automate as much as possible between checkpoints. Then set the time between checkpoints to be either a rapid cadence or even completely scriptable. Rinse and repeat.

10.7. Conclusion

Testing and validation of new, complex chipset architectures are of utmost importance for circuit designers and manufacturers beforehand to avoid hidden problems and to achieve the expected level of reliability of the consumer final product in fast-moving, high-competition, low-cost markets. We have considered validation and testing techniques, based on both physical measurements and model-based testing, which operate in the accuracy- and fault coverage- driven corners. Overview of both families of techniques has been presented, followed by their comparison and different synergetic applications. After examining how model-based testing can be exploited in combination with electrical physical measurements to enhance fault coverage, we consider a multi-domain combination of methods where model-based and physical methods generate and share the stimulus to perform spectrum analysis. The very high complexity of the circuits and the new trends, i.e., the multi-domain, multi-functionality nature of circuit chips, the exploration of different technologies and package solutions, and the convergence to integrated architecture will likely require an even greater convergence of modeling and application of fast experimental methods. We then consider how current modeling and focus on the quality in testing goods and services transition into the concept of zero service defect in the integrated track of the new harmonization and commenting on the inclusion of test and validation in the Life-Cycle Communication Packaging, which seeks to provide the supply chain with a tool to assist in creating appropriate communication packaging. The sixth section tackles the synergy status of reliability and test engineering, which is of primary importance for the life cycle of customers and users. Meeting customers' needs means making products that work properly, or provide them with the information to allow them to understand that a product is meant for a different use. Advancing quality management thinking, customers and users cannot obtain what they want without being provided with clearly written specifications.

10.7.1. Future Trends

The semiconductor industry strongly drives the general development of technology. As a consequence, there are always new requirements for semiconductor products to guarantee even more performances, more functionalities, and lower costs. This has important implications that reflect on product testing as well. The increase in product integration impacts the effort and the cost of product validation and test because the complexity of the design and the complexity of the circuitry and of the algorithms used within the product entail an intricate relationship that is hard to manage in practice. Complexity requires more effort and more care in the early phases of design validation, in order to keep the debug effort within reasonable limits. At the same time, since semiconductors are still subject to continuous cost reduction to maintain the overall

manufacturing cost for semiconductor products within acceptable limits in the consumer market, the implementation of design and production tests still needs to minimize test cost, usually at the expense of coverage and quality specifications. In the same way, the increase in product integration defines a strong trade-off between production cost and production yield, strongly influencing product lifetime quality. In practice, the embedded testing capabilities are dictated by the ability to manage the complexity at both design and system levels. In hardware, this refers to the analysis of the overall design architecture with respect to functional and structural testing architecture. In software, this refers to the analysis of the algorithms and the execution paths to define the code and access patterns that need to be validated and controlled.

References

Larsson, E. G., Edfors, O., Tufvesson, F., & Marzetta, T. L. (2014). Massive MIMO for next generation wireless systems. IEEE Communications Magazine, 52(2), 186–195.

Heath, R. W., Gonzalez-Prelcic, N., Rangan, S., Rappaport, T. S., & Murdock, J. N. (2016). An overview of signal processing techniques for millimeter wave MIMO systems. IEEE Journal of Selected Topics in Signal Processing, 10(3), 436–453.

Goldsmith, A. (2005). Wireless Communications. Cambridge University Press.

Tse, D., & Viswanath, P. (2005). Fundamentals of Wireless Communication. Cambridge University Press.

Biglieri, E. (2007). MIMO Wireless Communications. Cambridge University Press.