

# High Quality System Level Test and Diagnosis

Jutman, Artur; Sonza Reorda, Matteo; Wunderlich, Hans-Joachim

Proceedings of the 23rd IEEE Asian Test Symposium (ATS'14) Hangzhou, China, 16-19 November 2014

doi: <http://dx.doi.org/10.1109/ATS.2014.62>

**Abstract:** This survey introduces into the common practices, current challenges and advanced techniques of high quality system level test and diagnosis. Specialized techniques and industrial standards of testing complex boards are introduced. The reuse for system test of design for test structures and test data developed at chip level is discussed, including the limitations and research challenges. Structural test methods have to be complemented by functional test methods. State-of-the-art and leading edge research for functional testing will be covered.

Preprint

## General Copyright Notice

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

This is the author's "personal copy" of the final, accepted version of the paper published by IEEE.<sup>1</sup>

---

<sup>1</sup> **IEEE COPYRIGHT NOTICE**

©2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

# High Quality System Level Test and Diagnosis

Artur Jutman

Testonica Lab, Tallinn, Estonia

Matteo Sonza Reorda

Politecnico di Torino, Italy

Hans-Joachim Wunderlich

University of Stuttgart, Germany

**Abstract** - This survey introduces into the common practices, current challenges and advanced techniques of high quality system level test and diagnosis. Specialized techniques and industrial standards of testing complex boards are introduced. The reuse for system test of design for test structures and test data developed at chip level is discussed, including the limitations and research challenges. Structural test methods have to be complemented by functional test methods. State-of-the-art and leading edge research for functional testing will be covered.

**Keywords:** System test, board test, diagnosis

## I. INTRODUCTION

### A. Systems and System Testing

Many different objects are called a system, which - at a first glance - do not have much in common. A small smartphone in a pocket is a system as well as a telecommunication switch, which may fill a rack or even a complete room. A car may contain more than one hundred electronic control units (ECUs). Embedded systems may be implemented as a single system-on-a-chip, or such a system is seen as a complex board or even a rack of boards.

All of them have in common that they rely on complex interaction between hardware and software and communicate to the real physical world [1]. The term cyber-physical system combines information processing, sensing and acting with the outside environment, and it is often used as a substitute for embedded systems.

### B. Challenges in System Testing

System test has to be done not just after manufacturing, but also for system validation and during the rest of the lifecycle for maintenance and diagnosing field returns. The complexity of systems requires and limits the application of a divide-and-conquer approach at the same time. Test and diagnosis of all the components is mandatory, so-called Known Good Dies (KGD) are used for expensive packages [2], but they are by no means sufficient as their interplay has to be validated as well. Moreover, the integration and interaction of systems with the real world require the test of non-functional properties like power consumption, heat development and robustness against environmental conditions.

### C. System Validation

To speed up design and development, we aim at systems, which are correct at the first pass. However, with increasing scaling this goal is harder to reach as systems

get less predictable, and so-called post-silicon validation is becoming an accepted method [3], [4]. In this aspect, semiconductor testing gets similar to board testing, and appropriate techniques have to be adopted. Non-functional properties form an important aspect of system validation.

### D. Manufacturing test and diagnosis

Depending on the underlying technologies, boards may be repairable. In this case, also the defective device has to be identified, and diagnosis is essential for both repair and manufacturing improvement [5]. If repair is not possible in advanced technologies, diagnosis is still required for yield learning. For the same reasons, systems-on-a-chip are also subject to failure diagnosis [6].

### E. Diagnosis for maintenance and field returns

The failure rate in the field is a measure for the dependability and reliability of systems, and well-specified targets have to be kept or improved. Diagnosis is mandatory, and in application fields like automotive or avionics the root cause of any system failure has to be clarified [7]. For this purpose, two difficulties have to be overcome: First, many failures are only observable under specific operation conditions, and they may disappear after disassembling the system. These “No Failure Found” (NFF) cases are expensive and introduce also risk for other products [8]. Second, the OEM (Original Equipment Manufacturer), let us say a car manufacturer, relies on rather a long supply chain, and fault identification requires the collaboration of many partners (Fig. 1a). If boards and chips offer sufficient self-diagnosis capabilities already to the OEM, it will help to shorten the diagnosis process (Fig. 1b), and it can be used under typical operating conditions to reduce the NFFs before disassembling the system. The collected data will benefit not just the OEM but also all the members of the supply chain [9].

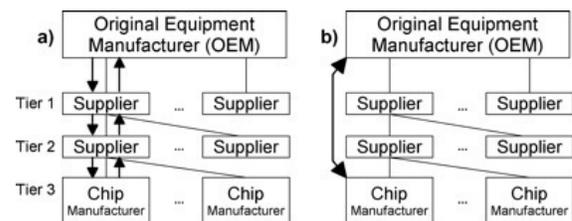


Figure 1. a) System diagnosis along the supply chain  
b) Built-in diagnosis

Test data can also be collected in the field on-line and off-line or concurrently and non-concurrently. The test

responses logged in the field will help to diagnose the field returns. Moreover, this type of test is mandatory to check the functionality of safety-critical systems.

#### F. Overview

The rest of this paper is organized as follows. The next section presents the common practices, current challenges and advanced techniques for board level test. Section 3 discusses the reuse of design-for-test infrastructure and test data developed for semiconductor manufacturing testing at system level. In section 4, the role of functional approaches in system test is explained. Some conclusions are finally drawn in section 5.

### II. BOARD-LEVEL TEST: COMMON PRACTICES, CURRENT CHALLENGES AND ADVANCED TECHNIQUES

End-of-line manufacturing test of board assemblies (PCBA) is one of the final test procedures before packing and delivering the product to the user. Typically, each produced board assembly has to pass a few different test phases before being qualified for shipping. The amount of these phases could be numerous for a modern complex electronic product and depends on its complexity and quality/reliability requirements. The particular combination of test techniques is also dictated by the economic feasibility [10][11], as each particular test type fits best for only a limited target class of defects, while covering additional defect classes is either impossible or requires extra costly effort. Hence, an efficient test strategy for a complex digital or mixed-signal product typically includes at least one technique per each category shown in Table I.

TABLE I. MAIN CATEGORIES OF PCBA TEST

<b>Inspection</b>	<i>Pre-reflow:</i> Solder Paste Inspection (SPI), Automated Optical Inspection (AOI) <i>Post-reflow:</i> Visual inspection, Automated X-ray Inspection (AXI), AOI
<b>Electrical test</b>	In-Circuit Test (ICT), Manufacturing Defect Analysis (MDA), Flying Probe Test (FPT)
<b>Scan test</b>	Boundary Scan (BS) and other test techniques based on IEEE 1149.1 and related standards (see Table 1)
<b>High-speed test &amp; measurement</b>	Processor-centric automated test solutions, FPGA-centric automated test solutions, Bit-Error Rate Test (BERT)
<b>Embedded instrumentation</b>	BIST instrumentation (fixed hardware), Synthetic instrumentation on FPGA (flexible hardware)
<b>Functional test</b>	Test of interfaces and basic behavior, test of main functions (fit-for-function test)

#### A. Classical PCBA test techniques

Inspection techniques help to check the general integrity of board assemblies including component presence, polarity, soldering quality, lifted leads, etc. Electrical test and measurement techniques are efficient when testing passive or analog components on the board by measuring their values, polarity, or parameters. The main challenge in

the mentioned two categories is designing faster and more accurate equipment, which in its turn is majorly a task of the mechanical engineering, material science and physics, thus being out of scope of the current paper.

Scan test (such as JTAG / Boundary Scan) [12] today is the industry's standard in board-level test providing an inexpensive yet efficient in terms of coverage and troubleshooting capabilities test technology. According to the 2009 iNEMI's study [13], about 80% of board test engineers see either high or moderate importance of accommodating Boundary Scan Test (BST) into the product test strategy. Taking into account a certain overhead in silicon area and pin count, a surprising 66% of product engineers reported either reduction or zero-influence of implementing BST infrastructure on the overall cost of the product development. Boundary Scan as well as several other scan-based techniques are governed by respective IEEE standards, which is one of the key factors in their widespread adoption and cost efficiency. A brief summary of the standards from the Boundary Scan family and their target application purpose is given in Table II, with the IEEE 1149.1 [14] being the forefather of the whole family and providing the basic architectural concept and test access principles. The latest version of IEEE 1149.1 was issued in 2013 [15] with major updates incorporated, including standardized means to control embedded instruments and pin-level electrical signal conditioning.

TABLE II. IEEE SCAN-BASED TEST ACCESS STANDARDS

Main Target Application	Main Purpose	Essential Technology	Target Fault Classes
<b>IEEE 1149.1 – Boundary Scan [14]</b>			
Manufacturing test of PCBA	Test access (TA) improvement	On-chip scan registers	Pin-level faults; net integrity
<b>IEEE 1149.4 – Mixed-Signal Test Bus [54]</b>			
Measurement: analog values	TA improvement	On-chip switches	Parametric values
<b>IEEE 1149.6 – BST of Advanced Digital Networks [55]</b>			
Testing LVDS high-speed nets	Test trough AC-coupled nets	On-chip pulse generators	Net integrity
<b>IEEE 1149.7 – Reduced-pin and Enhanced TAP [56]</b>			
Board test; SW debug	Flexible 2-pin high-speed TA	SERDES, addressing	Same as all above
<b>IEEE 1149.8.1 – Pin Toggle and Contactless Sensing [57]</b>			
Interconnect test of PCBA	Links to passive components	Capacitive sense plate	Net opens: AC and DC
<b>IEEE P1149.10 – High Speed Test Access Port (TAP) [58]</b>			
All of the above	High-speed test: data exchange	Reuse of high speed I/O pins	Same as all above
<b>IEEE 1500 – Embedded Core Test [59]</b>			
SoC-level test; IP core test	TA to IP cores in a SoC	Core wrappers	Digital domain faults inside IC
<b>IEEE 1687 – Embedded Instrumentation Access [60]</b>			
IC test, debug, diagnosis	Instrument access standard	Reconfigurable scan chains	Instrument-specific
<b>IEEE P1838 – Test Access for 3D Stacked ICs [61]</b>			
Test of 3DSiC integration	TA to through-silicon vias-TSV	Same as 1500, 1149.1, 1687	TSV integrity (mainly opens)

### B. Advanced and emerging PCBA test techniques

The major limitation of the classical BST is the inability to apply test patterns at-speed, hence limiting the covered fault spectrum to static (DC domain) faults. While the industry’s classical work-around has always been the usage of carefully crafted functional tests, the leading companies are adopting recently emerged high-speed or at-speed test techniques based on the automated (re-) configuration or programming of on-board programmable devices like FPGAs (referred to as FPGA-centric [16] or FPGA-controlled [17] test) and processors (referred to as processor-emulation [18], processor-centric [19] or processor-controlled [20] test). These techniques rely on JTAG infrastructure for test flow control while converting available on-board FPGA/CPU devices into embedded testers. Apart of the ability to cover timing-related faults (AC domain, delays, crosstalk, terminations), these techniques provide a very good test access degree due to the fact that FPGAs/CPU are typically backbone components of complex digital and mixed-signal devices by design [21]. When test is done, the test configuration is erased and the board is configured into its normal functional mode. Hence, no extra DFT overhead is needed. Today, only a few leading JTAG companies offer fully-automated tools that support this class of tests as a part of their software packages.

Another large class of the emerging board-level test techniques, whose adoption today is still in its infancy, is the *embedded instrumentation* [22]. In context of PCBA test, two major sub-classes of embedded instruments could be named: a) fixed built-in embedded circuits mainly in ASICs; b) synthetic reconfigurable multi-purpose instruments mainly in FPGAs. Typical examples of the former class are Memory BIST or PRPGs and error counters for Bit-Error Rate Test (BERT) of a communication channel. The lack of standardization and common practices limits wide adoption and reuse of such fixed embedded instruments at the board level, although IC-level applications of various BIST solutions are blossoming. On the contrary, the FPGA-centric synthetic embedded instrumentation is a very promising emerging board-level test technique [21].

Being the central part of a board and allowing fully flexible reconfiguration and reuse, the FPGA becomes an excellent embedded tester. A few cutting edge JTAG-based commercial test systems provide synthetic embedded instrumentation platform for the following applications:

- Memory test and BIST (on board);
- Bit-Error Rate Test (BERT) on communication channels (gigabit links);
- Test of common buses (LAN, SATA, PCIe, USB, CAN, LIN, I2C, SPI, etc.) and UART;
- In-system test and programming of non-volatile memories (flash devices);
- User-defined instruments.

Embedded instrumentation opens unprecedented potential in diagnostic access, monitoring and high-speed test. Studies show that industrial expectation towards bene-

fits of adoption of embedded instrumentation is currently very high [23]. Active industrial research in this area is very active with two main focus points: a) automation [16]; b) fault coverage improvement [24]. The new IEEE 1687 – IJTAG standard opens up the door towards seamless integration of tools, algorithms, instruments, IP cores and test patterns [25].

### C. PCBA-level fault models and testability metrics

There are several distinctive views on defect categorization, enumeration and coverage measurement at the board-level, as shown in Table III.

TABLE III. DIFFERENT APPROACHES TO TEST COVERAGE MEASUREMENT

Approach to fault modeling	Level of Abstraction	Examples of Defects	Test Coverage Metrics
Targeting defects in material and defects caused by assembly process	Structural faults at physical level	Bad soldering, lifted/bent leads, bad component, misalignment, tombstone, etc.	PPVS [63], MPS [64], PCOLA/SOQ [62]
Targeting pin-level and net-level defects	Structural and behavioral faults at logic level	Opens, shorts, bad driver (pin logic / buffer)	stuck-at for opens; zero, one and net dominance for shorts; stuck-driving and -not driving for pins [65]
Functional problems caused by defects	System level malfunction (behavioral)	Booting failure, unstable operation,	Functional model based test coverage metrics
Performance-related faults mainly at interconnect lines, buses, interfaces, communication links	Mainly statistical (error rates); structural approaches are missing but needed	High error rate (slow performance) crosstalk, jitter, delay fault	Bit error rates at communication links, but no universal industry-wide structural fault coverage metric [60]

While modeling static (DC) faults (first two rows in table III) has long ago become the industry’s standard with minor updates following progress in mounting/integration as well as test technologies, the AC domain (speed-related faults, last row in Table III) represents today a major research and standardization challenge. Except the BER measurement (that rather reflects the channel quality, i.e. signal/noise ratio, than presence of particular structural defects) there are no relevant industry-wide metrics used at

the board level to measure quality of high-speed or at-speed tests (e.g., run from embedded instrumentation).

The incompleteness of existing test coverage metrics has an important implication in terms of potentially missing test coverage due to incompleteness of test pattern sets as a result of the inability to adequately measure achieved test coverage with existing at-speed test set. In its turn, the potentially unknown lack of test coverage contributes to important No-Failure Found (NFF) problem [26], which is very costly. Hence, defect characterization and fault coverage metrics improvement is clearly a topic for extensive research.

### III. STRUCTURAL METHODS FOR SYSTEM-LEVEL TEST

#### A. Basics of Structural Testing

Even for a very small circuit it is impossible to verify its functionality completely as this task would require exponential effort in terms of inputs and flipflops. The problem is aggravated, if the interaction with the physical world, analog and mixed-signal circuits are included. A structural model may help to reduce the complexity significantly. The elements of structural testing are:

1. A model of the circuit structure (most often gate level).
2. A structural fault model like stuck-at, transition, delay or bridging faults. A general fault model is the conditional stuck-at fault model, which allows describing nearly all realistic faulty behaviors [27].
3. Changes of the circuit structure by additional design-for-test circuitry, which may introduce test modes during operation.
4. Structural test patterns, which detect a given percentage of faults, i.e., reach the required fault coverage.

With these modifications and additions, test time and test data volume do not increase any more exponentially by the circuits size but just linearly. However, not all failures may be covered by structural test, since neither the circuit model nor the fault model may be sufficiently precise and the test mode may hide failures. For this reason, during manufacturing both structural and functional strategies are applied [28][29].

#### B. Reuse of Structural Semiconductor Test Schemes

Table II shows design for test standards; some of them are used for semiconductor testing, some for board testing, and some for both.

In addition, most circuits contain proprietary internal structures like multiple internal scan paths, circuitry for test data compression and test response compaction and autonomous BIST hardware even for random logic like STUMPS schemes [30]. Usually, high quality structural test pattern sets are applied. For debug, diagnosis and post-silicon validation, even more complex reconfigurable scan networks (RSN) are integrated as suggested in IEEE 1687. Their high complexity may require special tools for control and verification. It seems natural to reuse all the internal

DfT structures also for system testing [31][32]. Yet this strategy poses challenges and may also create problems with respect to the workflow, safety and security. While reusing structural circuit test schemes for end-of-line manufacturing test of boards targets mainly organization, management and workflow, the reuse in the field creates even more challenges we will discuss below.

##### 1) Access

A core or a semiconductor component may be equipped internally with scan hardware, and test data has to be delivered from the outside. While this is a standard task during manufacturing performed by automatic test equipment (ATE), within a system it has to be performed by specialized controllers. The mentioned FPGA-centric instrumentation is an attractive way for implementing the flexible test controllers as well.

In addition, there may be no special media for the test data transfer and existing buses have to be reused. If a JTAG (IEEE 1149.1) bus is available, it is the obvious choice; however, sometimes it has to share the media with a functional bus. Fig. 2 shows an example how an SPI bus and JTAG share the same media to transport test data [33].

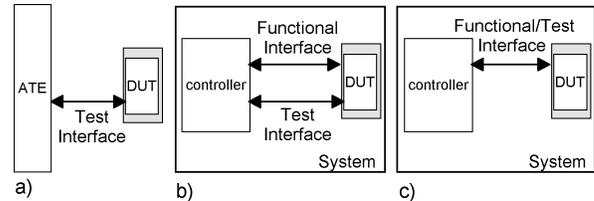


Figure 2. TAM reuse

##### 2) Dependability, safety and security

Usually, the DfT hardware at semiconductor level is not part of the specified and guaranteed properties to be used at higher levels or by customers. Depending on many circumstances, the DfT structure can be subject of changes not communicated outside. Moreover, documentation and quality levels of the infrastructure hardware may not follow the same rules as the rest of the circuit that is disclosed to the user. While it is obvious that test and diagnosis equipment and capabilities give some additional value to customers, their disclosure comes with additional costs and efforts. Here, management decisions have to be taken.

The disclosure of test and diagnosis infrastructure comes also with some risk. It must be excluded that the infrastructure interferes with system logic, and access has to be restricted. Security means have to be taken to guarantee authorized access [34]. Figure 3 shows different options to restrict the access to RSNs.

A restricting module RM grants access for certain features and test structures. Often, different levels of privileges have to be given depending on the authorization. For system validation, manufacturing failure analysis or a detailed inspection of field returns a complete access may be given. In a workshop during maintenance, access may be limited to collect just those data needed for system repair,

and the user may only be allowed to collect some Go/NoGo decisions.

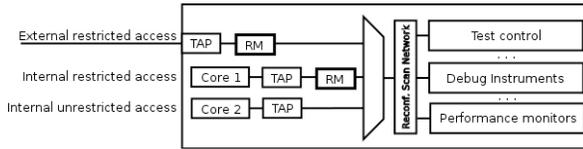


Figure 3. Restricting the access to the test infrastructure

The access policies do not only concern the interaction to the outside world, but also the interaction between components and cores. Some of them may be vulnerable or not trustworthy and their security may not be verified.

### 3) On-chip architecture for built-in diagnosis

As pointed out above, built-in self-diagnosis (BISD) allows us to search for faults and defects under the same environmental conditions as they appeared in the field without disassembling the system. However, BISD schemes developed for manufacturing testing are not well suited for system level diagnosis as they usually work in multiple phases. If a fault was detected during the self-test phase, additional runs for collecting intermediate diagnosis are executed in order to localize it [35][36].

If the STUMPS architecture is extended by a seed memory, a response memory for some intermediate signatures, and a fail memory which stores some failing intermediate signatures, a single pass test is sufficient (Fig. 4).

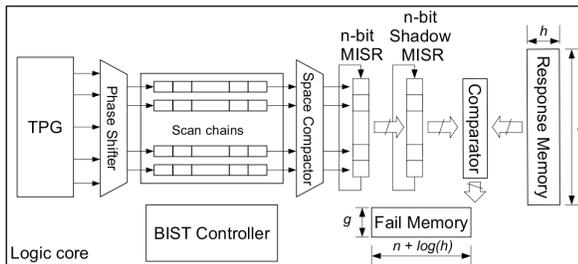


Figure 4. Scheme for single pass built-in self-diagnosis

The n-bit Multiple Input Signature Register (MISR) collects  $h$  intermediate signatures; each of them is passed to a shadow MISR that runs a few additional cycles in order to distribute also the last bits captured uniformly. The correct signatures are stored in the response memory and compared with the captured one. In case of a difference, up to  $g$  failing signatures including their indices are stored in the fail memory. Hence, we know up to  $g$  failing signatures, and that the signatures in between were correct. This information is sufficient to compute fault candidates with high resolution and accuracy [9].

Despite the additional costs, the added value of reusing structural test schemes from semiconductor manufacturing for system-level test and diagnosis is obvi-

ous. Yet like in production test, we cannot rely on structural test alone as discussed in the next section.

## IV. THE ROLE OF FUNCTIONAL TEST IN SYSTEM TEST

### A. Definitions

Different definitions exist for the concept of “Functional test”. In some case, a functional test is meant as a test, which does not rely on any Design for Testability structure: hence, this test only acts on the system functional inputs, and only observes the system functional outputs.

In other cases, a functional test is intended as a test, which has been generated by only exploiting functional information about the target system (i.e., without knowing its structure). As a consequence, this test does not rely on any structural fault model, leading to possible limitations in its defect coverage capabilities.

The two definitions can sometimes be adopted simultaneously: for example, a test can be generated starting only from functional information about the system, and the test is only applied resorting to functional inputs and outputs.

### B. Scenarios and motivation for functional test

Functional test may be adopted in different system test scenarios and may be motivated by different reasons. When addressing board-level test, functional test is typically considered as the final step (Table I), which is supposed to complement the previous ones with specific goals (e.g., testing the interfaces), allowing to achieve the target defect coverage. Further examples of usage of functional test at the system level include the following cases

- During the manufacturing test of a System on Chip (SoC), functional test may complement structural test because it may cover some defects that are not detected by the latter, e.g., because the former typically works at the system operational speed (while some DfT techniques do not), or because the functional test exercises the system exactly in the same conditions of the operational phase.
- Before mounting a device on a board, it may be required by regulations or economically convenient to perform a test to check whether the device is fault free (independently on the test performed by the device manufacturer). This test (sometimes called Incoming Inspection) is performed by the system company and it is often based on a functional approach, only (typically because possible Design for Testability features are not documented by the device provider).
- During the in-field test of a board, it may happen that the DfT features of the composing devices are not accessible any more (e.g., because they require an ATE), or are not documented by the device providers. Hence, the only feasible solution for the OEM company is often based on a functional test.

If the device provider does not deliver a proper test, the functional test has to be developed starting from the functions performed by each device, only.

### C. Functional test principles

In most cases, a system functional test requires a suitable test program TP to be executed by the processor(s) inside the system; this test program is expected to produce different results when the system is affected by a fault; results may be observed on a suitable output port or correspond to values left in a specified area of memory. When peripheral modules are targeted, suitable data stimuli TD may be required to be applied to specific inputs, or some output data have to be observed on specific output signals.

When functional test is the selected solution, two major issues have to be considered:

- How to apply the functional test, i.e., where to store the test program TP, how to trigger the processor to execute it, how to retrieve and check the produced results; a common solution lies first in storing the TP in an internal memory (or directly in the processor cache), triggering its execution through the interrupt signal, and then checking the results by accessing to the data memory (Software-Based Self-Test, or SBST) [38];
- How to generate the functional test (in particular, the test program TP).

Solutions to the first issue are typically dependent on the targeted system and to the existing constraints. Moreover, when in-field test is addressed, most of these tasks are commonly orchestrated by the Operating System.

### D. Functional test generation

Generating suitable test programs for functional test has been the subject of numerous research efforts, starting from [37], where the authors proposed a method to manually generate a test program for a simple processor, knowing its Instruction Set Architecture, only. Interestingly, the method was experimentally shown to be able to reach a good stuck-at fault coverage (around 90%).

In the last decades, the approach was extended to target processor cores of increasing complexity, as well as specific system components, such as memories, peripheral components and interconnection networks.

### E. Functional test generation for processors

A good overview of methods targeting processor cores is reported in [38]. More recently, researchers focused on specific modules within modern processor cores, such as Branch Prediction Units (BPUs), Memory Management Units (MMUs), Reorder Buffers (ROBs), and Cache controllers, showing that in most cases it is possible to develop test program that are guaranteed to reach a high fault coverage, without requiring the knowledge of the detailed implementation of such modules. Interestingly, some of the faults affecting these modules do not produce wrong results, but rather force the processor to behave in a temporarily different way, typically requiring a longer time

to complete the test program execution (performance faults). The detection of these faults may be particularly challenging, since it requires some techniques to observe the time behavior of the processor in a precise manner [39].

Recent efforts also targeted the development of functional test programs for multi-core processors [40] and GPUs [41].

The above methods mainly correspond to algorithms, allowing a skilled engineer to manually write a test program targeting a given module or a whole processor. However, the effort and time for achieving this result may be significant, and represents a major drawback of the functional approach. Previous efforts to automate the process, for example based on extensive simulation and evolutionary techniques [42], had a limited success, mainly due to the huge computational effort they require. Recently, it was shown in [43] that formal techniques can be successfully exploited to automatically generate functional test programs for a pipelined processor.

### F. Functional test generation for memories

Since memories correspond to an increasingly large fraction of a system, their test may represent an important target. Although the typical solution lies in the adoption of BIST, there are cases in which the functional approach is also of interest. In these cases the common solution lies in developing a test program, which performs on the target memory modules the same sequence of read and write operations mandated by a given March algorithm. In principle, this guarantees that the same defect coverage is achieved, although some defects may be missed due to the longer time between two consecutive accesses to memories [45]. An interesting extension of the same idea allows the test of cache memories resorting to suitably written test programs: [46] proposes a set of rules which allow to automatically transform any March algorithm into the corresponding test program. Reference [47] extends the same approach to L2 caches.

### G. Functional test generation for peripherals and interconnections

When targeting communication peripheral components, the functional approach requires the combined action of the processor, programming the component and exercising/observing it on one side, and that of an external body (e.g., an ATE), exercising/observing the component on the other side [66]. For in-field solutions, where the ATE can hardly be exploited, a loop-back connection is often adopted. A similar approach can be adopted for system peripherals, such as Interrupt and DMA controllers [44].

Several methods have been proposed to develop a functional test able to effectively detect faults in the interconnection structures within a system. As an example, the work in [48] targets structural faults in a Network on Chip.

#### H. Hot topics in functional test

In the last decade methods to generate functional test programs running under specific constraints (e.g., in terms of power [49]) or providing diagnostic information [50] were developed.

Both academia and industry are also exploring the cost and benefits stemming from the integration of the functional approach with a limited hardware supported, as proposed in [51] and [52]. When targeting board test, reuse of embedded instruments and on-board FPGAs (as introduced in Section III is also being explored.

Finally, it is worth mentioning that when more regular processor architectures are targeted, such as those of VLIW processors, it is possible to adopt a hierarchical approach, in which the global test program can be built, once the test program for each composing unit is known [53]. In this way the major drawback of the functional approach, corresponding to the huge cost for manually generating the test (as a consequence of the lack of automated tools), can be successfully faced.

Following this approach, new efforts are expected to be taken in the close future, aimed at automating the generation of functional test programs (to be adopted in different scenarios) exploiting some limited but well-defined information coming from the core or device producer, as well as from the system designer.

#### V. CONCLUSIONS

State-of-the-art and challenges of system level test open a wide and interesting research area, which covers nearly all aspects of testing today. This brief overview pointed out the main challenges of industrial practice and ongoing research today.

#### ACKNOWLEDGEMENTS

Parts of this work have been supported by the German Research Foundation (DFG) under grant WU 245/13-1 and WU 245/17-1, as well as by the European Commission through the FP7 STREP Project no. 619871 (BASTION) and European Regional Development Fund.

#### REFERENCES

- [1] J. Teich, "Hardware/Software Codesign: The past, the Present, and Predicting the Future," Proc. IEEE 100. Special Centennial Issue (2012): 1411-1430.
- [2] L. Gilg, "Known Good Die," Journal of Electronic Testing (JETTA) 10.1-2 (1997): 15-25.
- [3] S. Mitra, S. A. Seshia, and N. Nicolici, "Post-Silicon Validation Opportunities, Challenges and Recent Advances," Proc. ACM/IEEE Design Automation Conference (DAC'10), Anaheim, CA, USA, June 2010, pp. 12-17.
- [4] H. F. Ko et al., "Distributed Embedded Logic Analysis for Post-Silicon Validation of SOCs," Proc. IEEE Int. Test Conf. (ITC'08), Santa Clara, CA, USA, October 2008, pp. 1-10.
- [5] F. Ye, "Board-Level Functional Fault Diagnosis Using Artificial Neural Networks, Support-Vector Machines, and Weighted-Majority Voting," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 32.5 (2013): 723-736.
- [6] H. Tang et al., "Analyzing Volume Diagnosis Results with Statistical Learning for Yield Improvement," Proc. IEEE European Test Symposium (ETS'07), Freiburg, Germany, April 2007, pp. 145-150.
- [7] U. Abelein et al., "Non-Intrusive Integration of Advanced Diagnosis Features in Automotive E/E-Architectures", Proc. Design, Automation and Test in Europe (DATE), 2014
- [8] Z. Conroy et al., "A Practical Perspective on Reducing ASIC NTFs," Proc. IEEE International Test Conference, (ITC'05), Austin, TX, USA, November 2005, pp. 349-236.
- [9] A. Cook and H.-J. Wunderlich, "Diagnosis of Multiple Faults with Highly Compacted Test Responses", Proc. 19th IEEE European Test Symposium (ETS), 2014
- [10] B. Davis, The Economics of Automatic Testing, 2nd ed. McGraw-Hill, 1994, 416 p.
- [11] M. J. Smith, "The Real Cost of Not Testing!", Nordic Test Forum 2010, Nov. 23-24, Drammen, Norway.
- [12] K.P. Parker, The Boundary-Scan Handbook, Kluwer Academic Publishers, Boston, MA, USA, 2003, 373 p.
- [13] P.B. Geiger and S. Butkovich, "Boundary-Scan Adoption – An Industry Snapshot with Emphasis on the Semiconductor Industry", in Proc. Int. Test Conference (ITC'09), USA, 1-6 Nov. 2009, pp. 1-10.
- [14] IEEE Standard test access port and boundary-scan architecture, IEEE Std. 1149.1-2001 (R2008), IEEE 2002, 212 p.
- [15] IEEE Standard test access port and boundary-scan architecture, IEEE Std. 1149.1-2013. Working group URL: <http://grouper.ieee.org/groups/1149/1/>
- [16] I. Alekseyev, FPGA-Based Embedded Virtual Instrumentation, TUT Press, Tallinn, 2013, 155 p.
- [17] A. L. Crouch and S. A. Hack, "How P1687 Enables FPGA-Controlled Test", IEEE 10th International Board Test Workshop (BTW'2011), Fort Collins, USA, Oct 25-27, 2011.
- [18] H. Ehrenberg and T. Wenzel, Combining Boundary Scan and JTAG Emulation for advanced structural test and diagnostics. White Paper, GOEPEL electronics, 2009, 12 p.
- [19] A. Tsertov et al., "SoC and Board Modeling for Processor-Centric Board Testing", in Proc 14th Euromicro Conference on Digital System Design (DSD'2011), Oulu, Finland, Aug 31-Sept 02, 2011, pp 575-582.
- [20] J.A. Moore, "Processor-Controlled Test Enhances EMC's Test Effectiveness", IEEE 8th International Board Test Workshop (BTW'2009), Fort Collins, USA, Sept 15-17, 2009.
- [21] I. Alekseyev, A. Jutman, S. Devadze, S. Odintsov and T. Wenzel, "FPGA-Based Synthetic Instrumentation for Board Test," in Proc. of International Test Conference, Austin, USA, 2012.
- [22] How to test high-speed memory with non-intrusive embedded instruments. ASSET InterTech, Whitepaper, 2012.
- [23] Embedded Instrumentation: Its Importance and Adoption in the Test & Measurement Marketplace, Frost & Sullivan, Whitepaper, 2010, 20 p.
- [24] A. Jutman, "Fighting No Failure Found by Testing Dynamic Faults at Board Level", presented at Emerging Test Strategies - ETS2 of 19th IEEE European Test Symposium (ETS'2014), Paderborn, Germany, May 26-30, 2014.
- [25] A.L. Crouch, "IJTAG: The path to organized instrument connectivity", in Proc. International Test Conference, 2007, ITC 2007. pp. 1-10.
- [26] S. Davidson. "Towards an understanding of no trouble found devices," Proc. of VLSI Test Symposium, Palm Springs, California, USA, 2005, pp. 147-152.
- [27] S. Holst, S. and Wunderlich, H.-J.: "Adaptive Debug and Diagnosis Without Fault Dictionaries", Journal of Electronic Testing: Theory and Applications (JETTA), Vol. 25(4-5), August 2009, pp. 259-268
- [28] J. Zeng, et al., "On correlating structural tests with functional tests for speed binning of high performance design," Proc. IEEE Int.

- Test Conf. (ITC'04), Charlotte, NC, USA, October 2004, pp. 31-37.
- [29] P. Maxwell et al., "Comparing Functional and Structural Tests," Proc. IEEE Int. Test Conf. (ITC'00), Atlantic City, NJ, USA, October 2000, pp. 400-407.
- [30] P. H. Bardell et al., *Built in Test for VLSI: Pseudorandom Techniques*, John Wiley and Sons Ltd, 1987
- [31] T. Vo et al., "Design for Board and System Level Structural Test and Diagnosis," Proc. IEEE Int.l Test Conf., (ITC'06), Santa Clara, CA, USA, October 2005., pp. 1-10.
- [32] J. Qian, "Logic BIST Architecture for System-Level test and Diagnosis," Proc. IEEE Asian Test Symposium (ATS'09) Taichung, Taiwan, November 2009, pp. 21-26.
- [33] A. Cook et al., "Reuse of Structural Volume Test Methods for In-System Testing of Automotive ASICs," Proc. IEEE Asian Test Symposium (ATS'12), Niigata, Japan, November 2012, pp. 214-219.
- [34] R. Baranowski et al., "Securing Access to Reconfigurable Scan Networks", Proceedings of the 22nd IEEE Asian Test Symposium (ATS), 2013
- [35] W.-T. Cheng, M. Sharma, T. Rinderknecht, L. Lai, and C. Hill, "Signature based diagnosis for logic BIST," in Proceedings of the IEEE International Test Conference (ITC '06), pp. 1-9, 2006.
- [36] P. Wohl et al., "Effective diagnostics through interval unloads in a BIST environment," in Proceedings of the Design Automation Conference (DAC), pp. 249-254, 2002
- [37] S. Thatte and J. Abraham, "Test Generation for Microprocessors", IEEE Transactions on Computers, vol. 29, no. 6, pp. 429-441, June 1980.
- [38] M. Psarakis et al., "Microprocessor Software-Based Self-Testing", IEEE Design & Test of Computers, vol. 27, no. 3. May-June 2010, pp. 4-19.
- [39] M. Hatzimihail et al., "A methodology for detecting performance faults in microprocessors via performance monitoring hardware", Proc. of IEEE International Test Conference, 2007.
- [40] M. Kaliorakis et al., "Accelerated online error detection in many-core microprocessor architectures", IEEE 32nd VLSI Test Symposium (VTS), 2014.
- [41] S. Di Carlo et al., "A software-based self test of CUDA Fermi GPUs", IEEE 18th European Test Symposium (ETS), 2013.
- [42] F. Corno et al., "Automatic Test Program Generation: a Case Study", IEEE Design & Test of Computers, vol. 21, pp. 102-109, 2004.
- [43] A. Riefert et al., "An effective approach to automatic functional processor test generation for small-delay faults", Proceedings of the Conference on Design, Automation and Test in Europe (DATE), 2014.
- [44] M. Grosso et al., "Software-Based Testing for System Peripherals", Journal of Electronic Testing (JETTA), vol. 28 n. 2, pp. 189-200, 2012.
- [45] A. J. van de Goor et al., "Memory testing with a RISC microcontroller", Proc. of Design, Automation and Test in Europe (DATE), 2010.
- [46] S. Di Carlo et al., "Software-Based Self-Test of Set Associative Cache Memories", IEEE Trans. on Computers, vol. 60, issue 7, pp. 1030-1044, July 2011.
- [47] M. Riga et al., "On the functional test of L2 caches", IEEE International On-Line Test Symposium (IOLTS), 2012.
- [48] A. Dalirsani et al., "Structural Software-Based Self-Test of Network-on-Chip", IEEE 32nd VLSI Test Symposium (VTS), 2014.
- [49] J. Zhou and H.-J. Wunderlich, "Software-Based Self-Test of Processors under Power Constraints", Proceedings of the Conference on Design, Automation and Test in Europe (DATE), pp. 430-436, 2006.
- [50] P. Bernardi et al., "An Effective technique for the Automatic Generation of Diagnosis-oriented Programs for Processor Cores", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, pp. 570-574, 2008.
- [51] P. Bernardi et al., "Exploiting an infrastructure-intellectual property for systems-on-chip test, diagnosis and silicon debug", IET Computers & Digital Techniques, vol. 4(2), pp. 104-113, 2010.
- [52] F. Reimann et al., "Advanced Diagnosis: SBST and BIST Integration in Automotive E/E Architectures", Proc. 51st ACM/IEEE Design Automation Conference (DAC), 2014.
- [53] D. Sabena et al., "On the Automatic Generation of Optimized Software-Based Self-Test Programs for VLIW Processors", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, n. 4, pp. 813-823, 2014.
- [54] IEEE Standard for a Mixed-Signal Test Bus, IEEE Std. 1149.4-2010, IEEE 2011. Working group URL: <http://grouper.ieee.org/groups/1149/4/>
- [55] IEEE Standard for Boundary-Scan Testing of Advanced Digital Networks, IEEE Std. 1149.6-2003, IEEE 2013. Working group URL: <http://grouper.ieee.org/groups/1149/6/>
- [56] IEEE Standard for Reduced-pin and Enhanced-functionality Test Access Port and Boundary Scan Architecture, IEEE Std 1149.7, 2009. Working group URL: <http://grouper.ieee.org/groups/1149/7/>
- [57] IEEE Standard for Boundary-Scan-Based Stimulus of Interconnections to Passive and/or Active Components, IEEE Std 1149.8.1-2012, IEEE 2012. Working group URL: <http://grouper.ieee.org/groups/1149/atoggle/>
- [58] High Speed Test Access Port and On-chip Distribution Architecture, IEEE P1149.10, Working group URL: <http://grouper.ieee.org/groups/1149/10/>
- [59] IEEE Standard Testability Method for Embedded Core-based Integrated Circuits, IEEE Std 1500-2005, IEEE 2005. Working group URL: <http://grouper.ieee.org/groups/1500/>
- [60] IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device, IEEE Std 1687, IEEE 2014. Working group URL: <http://grouper.ieee.org/groups/1687/>
- [61] IEEE Standard for Test Access Architecture for Three-Dimensional Stacked Integrated Circuits, IEEE P1838, Working group URL: <http://grouper.ieee.org/groups/3Dtest/>
- [62] T. Taylor, "Functional Test Coverage Assessment Project", IEEE 8th International Board Test Workshop (BTW'2009), Fort Collins, USA, Sept 15-17, 2009.
- [63] C. Lotz, "LeanTest key: Test coverage analysis powered by traceability", IEEE 11th International Board Test Workshop (BTW'2012), Fort Collins, USA, Sept 11-11, 2012.
- [64] W. Rijckaert and F. de Jong, "Board Test Coverage – The value of prediction and how to compare numbers", in Proc. International Test Conference (ITC'2003), Charlotte, USA, Sept. 30-Oct. 2, 2003, pp. 190-199.
- [65] W. Feng et al., "Fault detection in a tristate system environment" in IEEE Micro vol. 21, no. 5, 2001, pp. 77-85.
- [66] A. Apostolakis et al., "Test Program Generation for Communication Peripherals in Processor-Based Systems-on-Chip", IEEE Design & Test of Computers, vol. 26 n. 2, pp. 52-63, 2009.