FP7-ICT-2013-11-619871

BASTION

Board and SoC Test Instrumentation for Ageing and No Failure Found

Instrument: Collaborative Project
Thematic Priority: Information and Communication Technologies

Prototype tools
(Deliverable D6.2)

Due date of deliverable: April 30, 2017
Ready for submission date: June 27, 2017

Start date of project: January 1, 2014 Duration: 40 months

Organisation name of lead contractor for this deliverable: Testonica Lab

Revision 1.0

Project co-funded by the European Commission within the Seventh Framework Programme (2014-2017)		
Dissemination Level		
PU	Public	☒
PP	Restricted to other programme participants (including the Commission Services)	☐
RE	Restricted to a group specified by the consortium (including the Commission Services)	☐
CO	Confidential, only for members of the consortium (including the Commission Services)	☐

Notices

For information, please contact Artur Jutman, e-mail: artur@testonica.com

This document is intended to fulfil the contractual obligations of the BASTION project concerning deliverable D6.2 described in contract 619871.

Table of Revisions

Version	Date	Description and reason	Author	Affected sections
0.1	April 18, 2017	Structure created	A. Jutman	All
0.2	April 27, 2017	Testonica tools added	S. Devadze	Section 2.4, Section 2.6, Section 2.7, Section 2.8, Section 2.9
0.3	May 3, 2017	PDT IJTAG test tool added	R. Cantoro	Section 2.3
0.4	May 8, 2017	IRF tools added	H. G. Kerkhoff	Section 2.10, Section 2.11
0.4a	May 9, 2017	IJTAG retargeting tool added	R. Krenz-Baath	Section 2.5
0.5	May 12, 2017	ASTER tools added	C. Lotz	Section 2.12, Section 2.13
0.6	May 14, 2017	Rejuvenation tool added	M. Jenihhin	Section 2.1
0.7	June 9, 2017	Hierarchical In-Field Test Solutions	P. Engelke	Section 2.2
0.8	June 12, 2017	Common demonstration platform description	A. Jutman	Section 3
0.9	June 21, 2017	Writing Introduction and Conclusions	A. Jutman	Section 1 and Section 4
0.9a	June 26, 2017	Correcting references and numbering	A. Jutman	All
1.0	June 27, 2017	Improvement of Section 3, final polishing and preparing for submission	A. Jutman	Section 1, Section 3, Section 4

Author, Beneficiary

Artur Jutman, Sergei Devadze, Testonica Lab
Christophe Lotz, ASTER Technologies
Piet Engelke, Infineon
Hans G. Kerkhoff, Hassan Ebrahimi, University of Twente
Maksim Jenihhin, Jaan Raik, TU Tallinn
Erik Larsson, University of Lund
Matteo Sonza-Reorda, Riccardo Cantoro, Politechnico di Torino
Rene Krenz-Baath, HSHL

Executive Summary

This document describes BASTION prototype tools aiming at demonstrating the full spectrum of RTD activities in the project, i.e. work packages WP1, WP2, WP3 and WP4. This report inherits from D6.1, which listed demonstration plans and indicated experiments to be demonstrated at the end of the project. First, individual prototype tools are presented, then the common demonstration platform is described, which is built around a complex board that represents the common test case (system under test) used throughout the whole demonstration process. This deliverable concludes WP6.

List of Abbreviations

AOI	- Automated Optical Inspection
ASIC	- Application-Specific Integrated Circuit
AXI	- Automated X-ray Inspection
BERT	- Bit Error Rate Testing
BBIST	- Board BIST
BIST	- Built-In Self-Test
BOM	- Bill of Material
BSDL	- Boundary-Scan Description Language
BST	- Boundary-Scan Test
CAD	- Computer Aided Design (also EDA)
CMOS	- Complementary Metal-Oxide Semiconductor
CPU	- Central Processing Unit, also Processor
DDR	- Double Data Rate
DPM	- Defects Per Million
DPMO	- Defects Per Million Opportunities
DRAM	- Dynamic RAM
DRC	- Design Rule Check
EDA	- Electronic Design Automation (also CAD)
EMS	- Enhanced Manufacturing Services
FIFO	- First In, First Out (data buffer)
FMC	- FPGA Mezzanine Connectors
FPGA	- Field Programmable Gate Array
FP7	- European Union's 7th Framework Program
FPT	- Flying Probe Test
FPY	- First Pass Yield
FT	- Functional Test
HDMI	- High-Definition Multimedia Interface
HW	- Hardware

IBIS	- Input Output Buffer Information Specification
IC	- Integrated Circuit
ICL	- Instrument Connectivity Language
ICT	- In-Circuit Test
IIC	- Inter-Integrated Circuit, also I^2C
IJTAG	- Internal JTAG, a short name for IEEE 1687 standard and infrastructure collectively
IP	- Intellectual Property (hardware module in FPGA or SoC)
IST	- Information Society Technologies
JTAG	- Joint Test Action Group; also Boundary Scan; often used as a short name of the IEEE 1149.1 standard and respective infrastructure including test access port and header on the board;
MPS	- (Coverage metrics based on) Material, Placement & Solder
NBTI	- Negative Bias Temperature Instability
NFF	- No Fault Found or No Failure Found (also NTF)
NoC	- Network-on-Chip
NTF	- No Trouble Found (also NFF)
OTG	- USB On-The-Go
PCOLA/SOQ	- (Coverage metrics based on) Presence, Correct, Orientation, Live, Alignment/Short, Open & Quality
PCBA	- Printed Circuit Board Assembly
PCI	- Peripheral Component Interconnect
PCIe	- PCI Express
PDL	- Procedural Description Language
PMOS	- p-type Metal Oxide Semiconductor
POST	- Power-On Self-Test
PPVS	- (Coverage metrics based on) Presence, Polarity, Value & Solder
PVT	- Process, Voltage, Temperature
RAM	- Random-Access Memory
RTD	- Research and Technological Development
SBST	- Software-Based Self-Test
SIB	- Segment Insertion Bit
SMA	- SubMiniature version A (connector)
SoC	- System on Chip
SODIM	- Small Outline Dual In-Line Memory
SFP+	- Enhanced Small Form-Factor Pluggable (transceiver)
SVF	- Serial Vector Format
SW	- Software
TDR	- Test Data Register
UART	- Universal Asynchronous Receiver/Transmitter

USB	- Universal Serial Bus
UUT	- Unit Under Test
VHDL	- VHSIC Hardware Description Language
VHSIC	- Very High Speed Integrated Circuit

Table of Contents

1 Introduction

The main target of the demonstration work package WP6 is twofold: to demonstrate key achievements in BASTION project and to allow combined usage of tools and complementary techniques. More specifically, the WP aims at:

- Identify a test case to be used throughout the whole project, combining the activities at core, IC and PCBA level and thus demonstrate the achievements that can be reached by the partners;
- Collect and report all the output data gathered by different partners in the final experiments performed within the WPs according to the plan in the previous item;
- Develop a plan for allowing the combined usage of tools developed within the project, so that they can be used in an integrated manner (see D6.1 and Section 3 of this deliverable);
- Put together a set of demonstrations with corresponding data that can be exploited by WP5 in order to better disseminate the results and achievements produced by the project.

A tremendous effort has been spent by the BASTION consortium on defining a common vision towards the demonstration platform and studying technical feasibility of this target as well as actual development work, which eventually has been successfully achieved.

BASTION project has resulted in an impressive number of prototype tools (13 items reported in Section 2 of this document) with at least half of them developed by industrial partners or in a direct collaboration with them. Many tools are prototypes at high technology readiness level: ready to be converted into commercial tools, which promises a successful exploitation and commercialization of BASTION project results while the demand for embedded instrumentation on the market and expectations are high [2].

The main experimental outcome of the joint collaboration of all BASTION partners in frames of WP6 is the ability to demonstrate by calculations how close we have approached our target of reducing test escapes and impact of NFF [1] by a factor of 2 (KPI5). Other experimental results have been reported within respective workpackages.

This document describes BASTION prototype tools aiming at demonstrating the full spectrum of RTD activities in the project, i.e. work packages WP1, WP2, WP3 and WP4. This report inherits from D6.1, which listed demonstration plans and indicated experiments to be demonstrated at the end of the project. First, individual prototype tools are presented, then the common demonstration platform is described, which is built around a complex board that represents the common test case (system under test and monitoring) used throughout the whole demonstration process.

1.1 Structure of the document

Individual prototype tools are described in Section 2. The common demonstration platform and respective storyline is presented in Section 3, which also summarizes the progress achieved on KPI5. Section 4 concludes the report.

2 Individual Prototype Tools

This section lists individual prototype tools while Section 3 will demonstrate the way they interact with each other (apart from several tools, which could not be integrated).

2.1 NBTI aging analysis and rejuvenation stimuli generation tool (zamiaCAD framework with uGP tool)

Related WP: WP1, WP3, WP4
Partners: TUT, PDT

This prototype tool demonstrates feasibility and efficiency of a novel approach proposed in BASTION to mitigate NBTI in processor circuits and stand-alone cores using rejuvenation of pMOS transistors along NBTI-critical paths with dedicated programs or binary stimuli. The flow is integrated into an open-source framework zamiaCAD (see Figure 1).

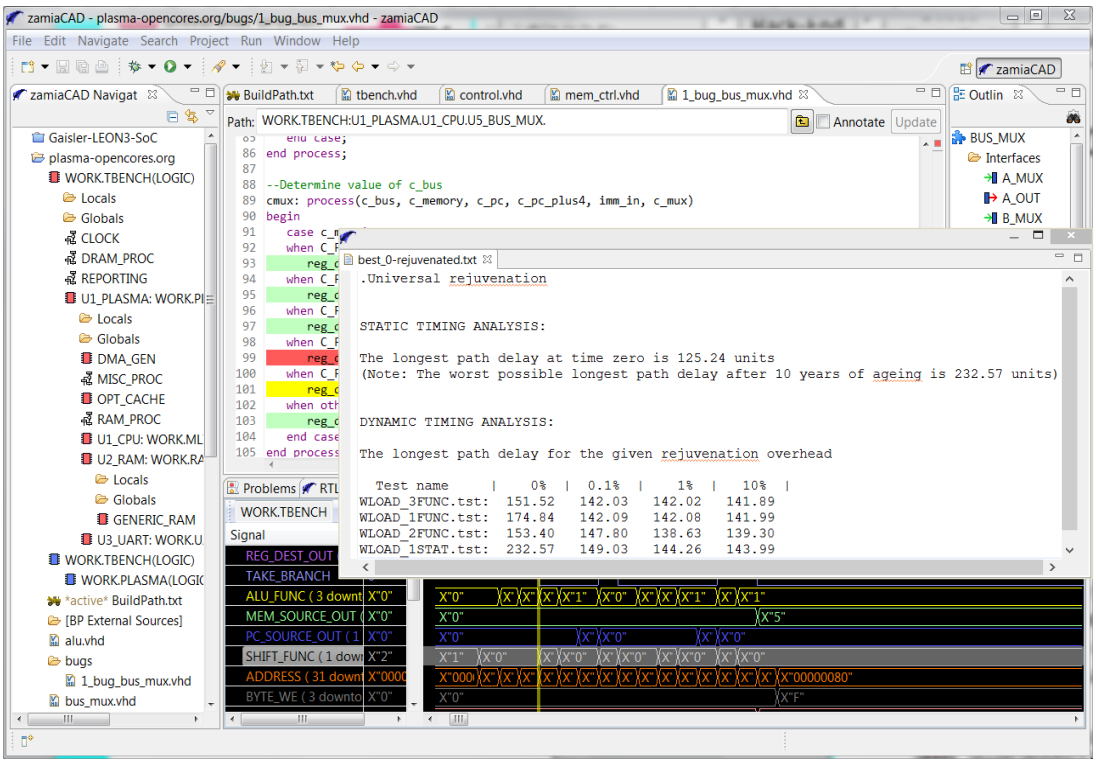


Figure 1. User interface of the open-source framework zamiaCAD

The method incorporates hierarchical fast, yet accurate identification of NBTI-critical paths at gate level and the rejuvenation Assembler programs generation using an Evolutionary Algorithm. In this flow (see Figure 2) we exploit a general-purpose evolutionary toolkit called *µGP* [4] and find a suitable fitness function using an open source hardware analysis framework called *zamiaCAD* [5]. The advantage of such flow lies in its flexibility for solving the dependencies of impacts by individual gates to the most critical NBTI-induced path delay by using evolutionary optimization processesThe generated rejuvenation programs are applied at predefined periods in order to drive transistors to the recovery phase and thus to extend the reliable lifetime

of any processor implemented in nanoelectronic technology. More details on the method are provided in deliverable D3.3

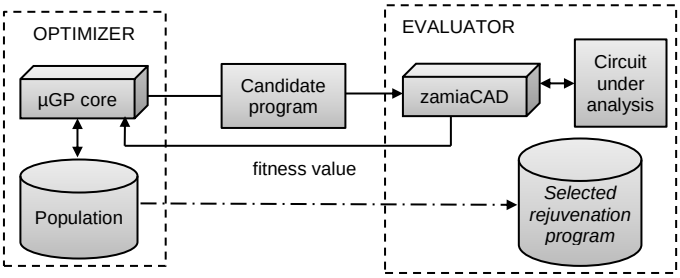


Figure 2. General flow of the evolutionary generation of rejuvenation programs

The open-source zamiaCAD framework is being developed and maintained at TUT. Its front-end includes a parser and an elaboration engine that both support full VHDL-2002 standard specification and a set of VHDL-2008 extensions. On the back-end side, the framework allows design simulation, static analysis and other applications for scalable hardware design and analysis. zamiaCAD has an Eclipse IDE plug-in based graphical user interface for advanced design entry and navigation. The BASTION NBTI analysis and rejuvenation flow is automated and configured by Python scripts. These also the µGP tool for evolutionary stimuli generation. µGP is a general-purpose evolutionary toolkit developed at PDT. It allows a high degree of customization of evolutionary operators, stop criteria, and algorithm parameters. Internally it represents candidate solutions as multi-graphs, where each node roughly corresponds to a locus of the genome. These approach has high efficiency for generating rejuvenation binary stimuli for stand-alone cores [13] and rejuvenation programs for processor circuits [14].

The NBTI ageing analysis and rejuvenation stimuli generation tool is publically available at the zamiaCAD website.

2.2 Architectures and Instrumentation for Hierarchical In-Field Test

Related WP: WP2, WP3, WP4
Partners: IFAG

Within the BASTION project IFAG contributed a multitude of implementation concepts for hierarchical in-field test. These concepts will also be made available in IFAG's in-house design flow. Typically there is a close interaction between an implementation concept and the EDA tools used in various phases of the development process of complex SoCs. Hence, in some cases modifications to the design flow will be required to enable realization of the concepts developed in BASTION. Usually this means updating templates and scripts controlling the different EDA tools. In case a dedicated IP is required to implement a certain concept, this is made available in IFAG's in-house database.

In the following we give a brief overview of the main implementation concepts elaborated in detail in several deliverables of the BASTION project. In D3.1, IFAG proposed a design measure to improve the localization of errors using LBIST. The

approach was validated experimentally on a small test-case design using a commercial EDA tool. The LBIST exploits an implementation concept for a module test isolation developed in D3.2. In that deliverable, the isolation concept is adapted to several applications scenarios, and experimental results obtained for two industrial cores are reported. In D3.3 implementation guidelines for hierarchical test were defined. Following these guidelines helps to make optimal use of module test isolation. Results were again validated for an open source quad-core processor using commercial EDA tools. In D4.2 we presented the concept of a BIST for SAR-ADCs. This concept, based on capacitance measurements, was realized in a circuitry that was implemented on a test-chip. Measurement results from experiments with this silicon implementation were reported in D2.2. In D2.1, IFAG presented an experimental case study for an IJTAG network that controls three of the aforementioned BIST instruments for SAR-ADCs. The target of that study was to specifically optimize this IJTAG network for in-field test. Focusing on re-using IJTAG for in-field test, D2.3 describes a concept to ensure safety and security of such networks. Again results were validated for a test-case design based on the open source quad core processor using commercial EDA tools.

2.3 IEEE 1687 IJTAG Network Testing Tool

Related WP: WP3
Partners: PDT, ULUND, TL

The tool deals with the test of permanent faults affecting an IJTAG network. In particular, the focus is given to the reconfigurable elements, such as SIBs and ScanMuxes. The test checks whether the network can be properly configured and whether it works as expected in that configuration (i.e., when a certain set of instruments can be accessed).

The tool, written in Java, is able to read an IJTAG network topology described in different formats including ICL. The ICL Tools Software library provided by Testonica has been included in the tool for this purpose. The IJTAG network is parsed and transformed into an internal format corresponding to a directed graph. The elements of the scan infrastructure (scan input/output, TDRs, SIBs, ScanMuxes) are associated to vertices, while the connections between elements are represented by edges. Edges are labeled with the configuration values of reconfigurable modules. For example, a SIB vertex has two outgoing edges, labeled asserted and de-asserted, respectively. Every possible path in the scan network is represented by a path in the graph from scan input to scan output vertices.

Several algorithms are used to traverse the graph and to produce test patterns [9][10]. The tool maintains a list of possible faults, that are associated to the elements of the scan infrastructure. Test algorithms move the network into certain configurations, in which test vectors are shifted in from the scan input, and the expected responses are observed on the scan output. Each time a test vector is applied, the fault list is updated. By carefully identifying target edges on the graph, the tool is able to generate test vectors until the full test coverage is reached. The generated patterns are exported to SVF.

In details, the tool performs the following operations:

- Reading the IJTAG network from ICL and checking for errors
- Traversing the network and building the internal data structure
- Identification of the fault list
- Test pattern generation by traversing the graph
- Generation of the test patterns in SVF.

2.4 IEEE 1687 IJTAG Infrastructure Validation Tool

Related WP: WP2
Partners: TL, TUT

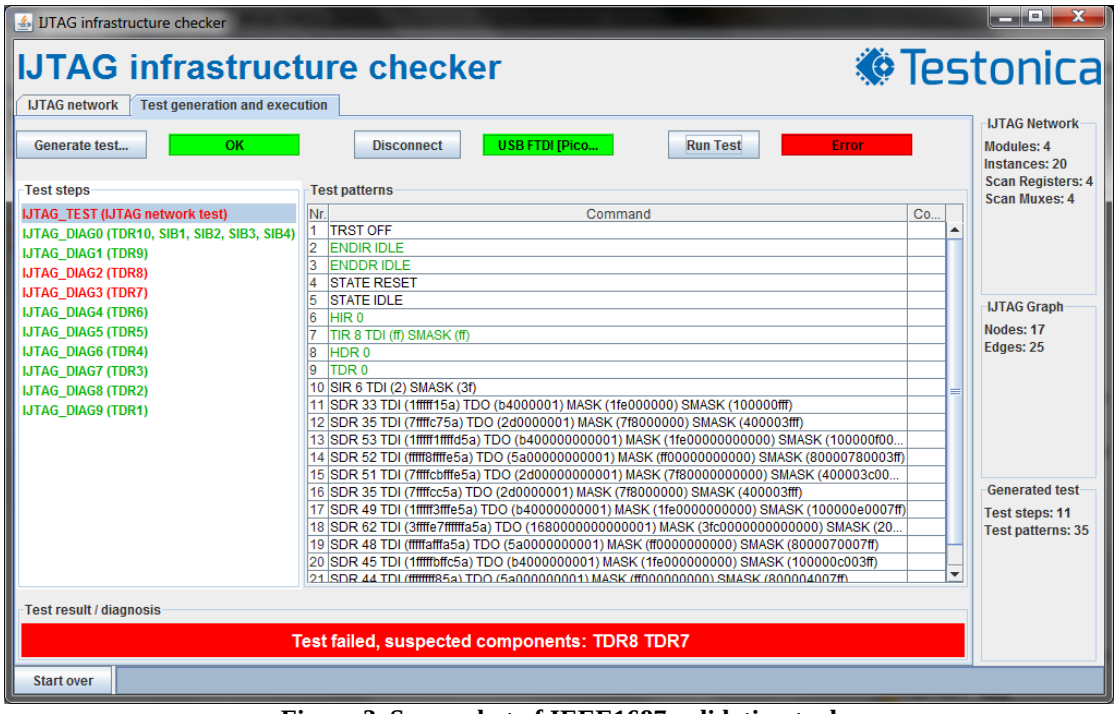


Figure 3. Screenshot of IEEE1687 validation tool

IEEE1687/IJTAG infrastructure validator is a prototype tool that is capable to analyze specification of IJTAG network (in IEEE1687 ICL format) and check whether the actual implementation of the network in hardware matches to this specification.

The validation begins with simple syntax and semantics checks of source ICL description and is followed by the examination of its compliance to the rules of IEEE 1687 standard. For hardware validation, the tool generates a set of diagnostic test patterns. These patterns used to ensure that all the segments of on-chip IEEE1687 network and all instruments in this network can be accessed in the way described in the specification. More details on the algorithm of IJTAG validation are provided in deliverable D2.3.
The prototype tool is capable to perform the validation on real hardware and also provide end-user with diagnostic information in case if mismatches between specification and implementation are detected.

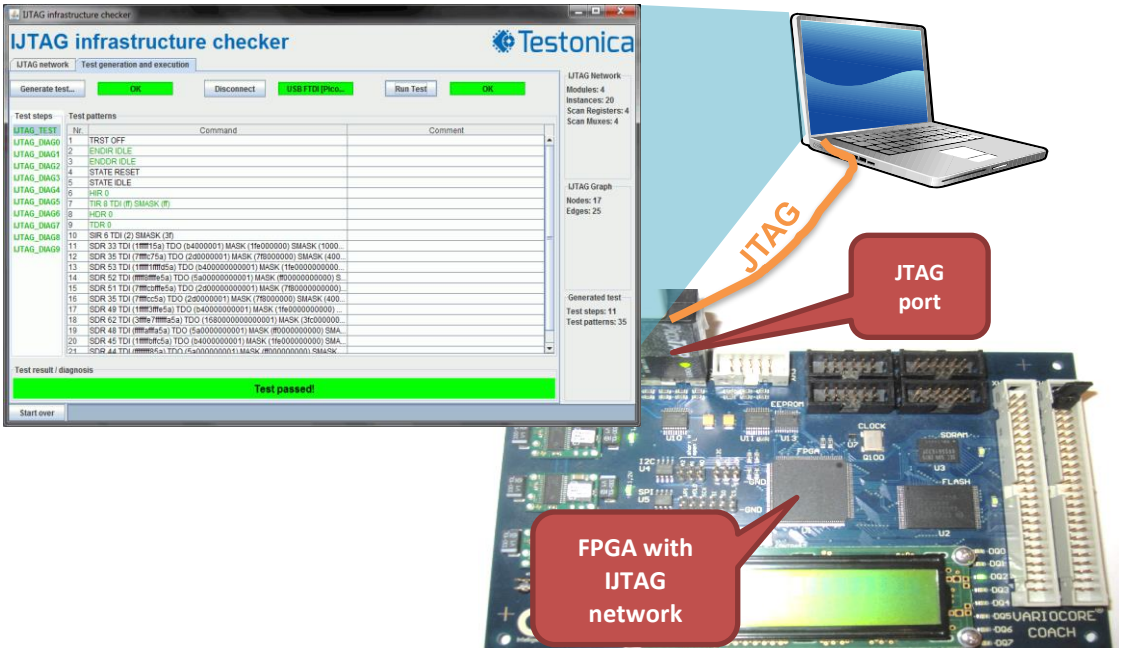


Figure 4. Experimental setup for IEEE 1687 IJTAG validator prototype

To verify the prototype tool we have implemented an IJTAG network in FPGA (8 instruments, 10 scan registers, 7 scan multiplexers). We also provided fault injection mechanism to emulate design errors (e.g. unconnected input of multiplexer, erroneously swapped network segments, wrong control condition of an multiplexer). The demonstrator was able to detect all injected faults and report diagnostic results.

2.5 Optimal PDL-Retargeting in IEEE 1687 IJTAG Networks

Related WP: WP2, WP3
Partners: HSHL, ULUND

This SW-demonstrator provides a solution for the important task of retargeting PDL-instructions on a given IJTAG network. In particular the tool (if required) is able to retarget a set of PDL-instructions within a single iApply-group such that number of shift cycles required to apply/execute the retargeted PDL instructions is proven to be minimal. Hence the actual application time of the test vectors is minimal too. Besides retargeting the tool can be applied to perform verification tasks on the IJTAG network such as reachability analysis, deadlock analysis as well as property checking tasks.

The tool reads the description of an IJTAG network provided in SMV-format. The instructions are described in PDL Level-0. Depending on a given option the tool generates test vectors in SVF-format or computes the upper-bound of the IJTAG network.

The upper-bound computation discussed in [12] derives an FSM-representation of the IJTAG network in order to analyse the longest possible path between two reachable states and, based on that result, derive the required number of time frames for the

sequential (0,1)-ILP problem. As mentioned in the paper the network is modified and decomposed to reduce the complexity of the FSM and hence increase the robustness of the tool to deal with a wide range of complex and large IJTAG networks.

As described in [11] the actual retargeting of the PDL-instructions is accomplished by firstly modeling a sequential (0,1)-ILP optimization problem from the internal representation of the IJTAG network, next computing an optimal solution by using Minisat+, and finally deriving the resulting test vectors from solution provided by the reasoning engine.

In detail, the tool performs the following operations:

- Reading the IJTAG network from SMV file
- Reading PDL-0 instructions from a PDL file
- Generating an internal data structure of the IJTAG network
- Computing the generalized upper-bound of the IJTAG network
- Computing test vectors to execute PDL-0 instructions on the IJTAG network
- Generation of the test patterns in SVF.

2.6 Fault Management Toolset

Related WP: WP3
Partners: TL, TUT, IFAG

This prototype toolset demonstrates capabilities of in-situ instrument-assisted fault management technique developed in WP3. More information about fault management is presented in deliverables D3.1, D3.2 and D3.3.

The developed fault manager prototype is targeted to run on embedded Linux platform (PetaLinux) hosted on two-core ARM Cortex A9 processor of the Zynq ZC706 Evaluation Kit [8] (see Section 3). The tool-set consist of several parts:

1) Kernel module that implements fault management features (reaction on faults detected by instruments, maintaining system health map, scheduling tasks according to health map information);

2) Client-side software (starting/stopping fault managed processes, reporting overall system health information);

3) IEEE 1687 instrumentation network with asynchronous signaling based on FCX status flags;

4) Concurrent checkers [3] embedded into NoC routers on FPGA and aging monitor (on the RELY IC [6]);

5) Instrument Manager component implemented on FPGA.

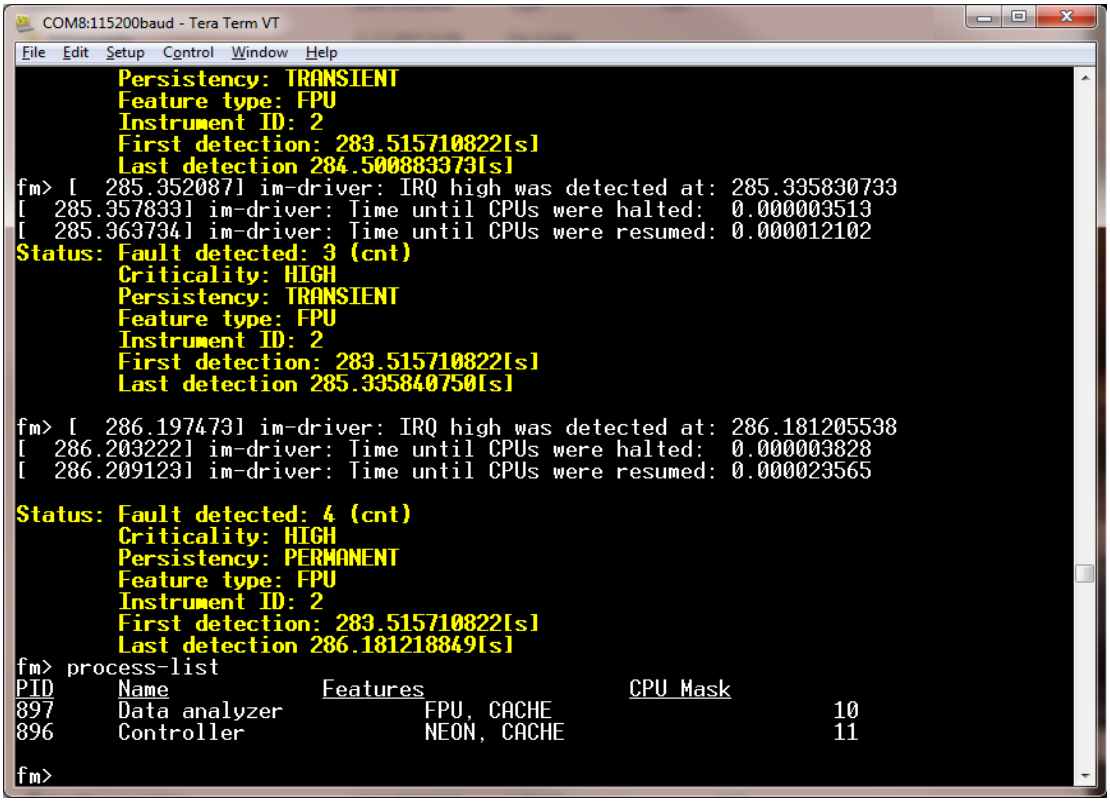


Figure 5. Fault management console integrated into embedded Linux OS

In the developed demonstrator it is possible to emulate faults in hardware by pressing an on-board switch on the ZC706 kit. When a “fault” occurs, fault manager is able to detect the fault event, localize the source of the fault, update health map and re-schedule the running processes in accordance with the updated health status of the system.

2.7 Power-on Board Self-Test (Board BIST)

Related WP: WP4
Partners: TL

This prototype tool demonstrates abilities of power-on Board BIST methods developed in Task 4.1. Board BIST technique helps to identify defects that can emerge or manifest themselves during in-field operation (for example, intermittent faults, marginal defects, damaged parts, wear-out, etc). The method is described in detail in deliverable D4.1

The prototype tool consists of a number of Phyton scripts. The script takes as an input a set of Boundary Scan (BS) interconnection test patterns generated by traditional BS test development software. Then, the tool automatically generates a HDL design that can be synthesized into BBIST firmware and put on target board.

The capabilities of BBIST are demonstrated on Zynq ZC706 FPGA development board (see Section 3). First, the interconnect test was generated for the board using BS software and exported into SVF format. In total, 41 test patterns have been generated. Then, based on the generated patterns BBIST IP has been synthesized and FPGA configuration bit-stream has been produced.

We have validated BBIST demonstrator by injecting different types of faults (opens and shorts) into various places of board interconnect. In case of injected fault, BBIST IP reported failure by flashing an on-board LED halting board booting process.

2.8 Embedded Instrument for Testing High-Speed Serial Communication Links

Related WP: WP4, WP3
Partners: TL

This prototype demonstrates ability of the developed embedded instruments to assess quality of a gigabit serial links and detect delays, marginal delay faults and other possible defects on the link.

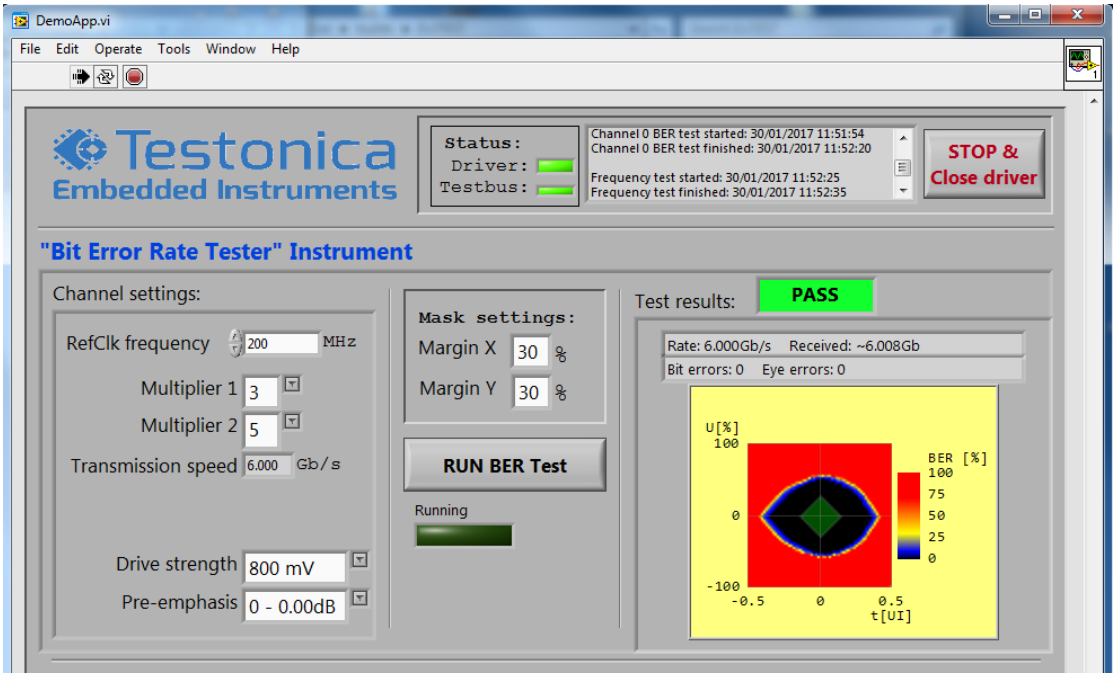


Figure 6. Screenshot of high-speed serial link tester prototype

The prototype tool is capable to run at-speed test sequences over gigabit serial link, detect errors occurred during test transmission and calculate bit-error rate (BER). But the most important feature of the instrument prototype is construction of so-called BER eye diagrams for assessing the quality of serial link.

By analyzing BER eye diagrams, it is possible to detect marginal faults, i.e. defects that do not lead to immediate failures in data transfer but still can exhibit themselves under certain circumstances. In case of such defect, the measured BER eye diagram becomes smaller (shrunk) in comparison with diagram of defect-free PCBA link.
The prototype has been implemented on Zynq ZC706 Evaluation Kit (see Section 3).

2.9 Marginal delay fault tester (prototype for DDR3 memories)

Related WP: WP4
Partners: TL

This prototype instrument is developed to test DDR3 interconnection lines for possible marginal delay faults. The prototype performs tests of memory interconnect with various types of test patterns and constructs so-called “bath-tub BER” diagrams for each data line. For this purpose, the instrument uses a special calibration mode of DDR memories (normally intended to fine-tune bus timings with respect to particular PCBA) for shifting sampling point on particular data line during data transfer. More details on the method are provided in deliverable D4.3

Figure 7. Test for marginal faults. Results for correct (left) and defective (right) DDR modules

The prototype has been validated in the following experimental environment. We have selected five SODIMM DDR3 memories of the same part-name (KVR16S11S6/2). Also, we injected a “defect” into one of the SODIMMs by isolating (taping) two ground pins on SODIMM module. It is important to stress, that even with the injected “defect”, the memory module is still operational and passes functional tests. By analyzing BER bath-tub diagrams measured with the developed prototype, it is possible to clearly see that signal quality has been significantly degraded on the corresponded data lines of “defected” module.

2.10 Intermittent Resistive Fault injector

Related WP: WP1, WP2, WP4
Partners: UT

An automated hardware/software platform for emulation-based IRF injection has been developed. The proposed emulation-based platform is composed of a host computer, an FPGA and an in-house PCB board which is shown in Figure 8.

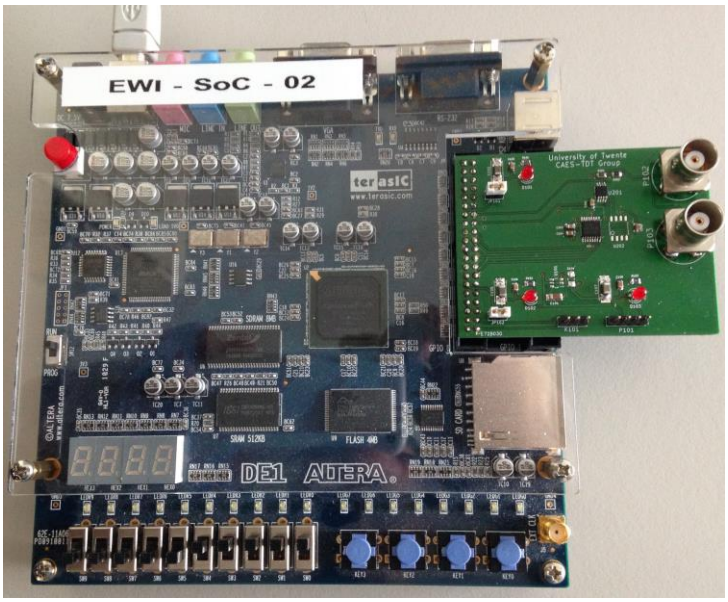


Figure 8. Actual implementation of the IRF generator. The FPGA houses the digital control, and the right-hand board the switches and resistors.

The host computer executes a Matlab script which generates a burst of random resistance values based on the model was described in deliverable D1.1. The communication screen on the computer, together with the essential parameters can be seen in Figure 9. The generated burst sequence can be transferred to the FPGA board by a serial communication link.

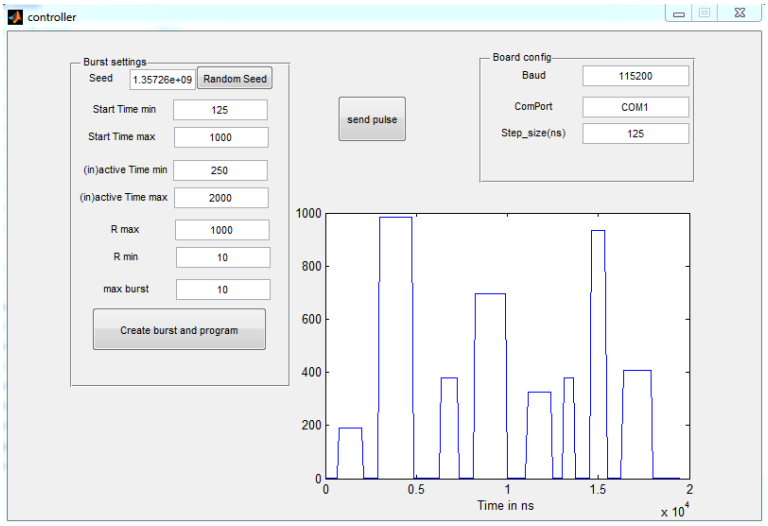


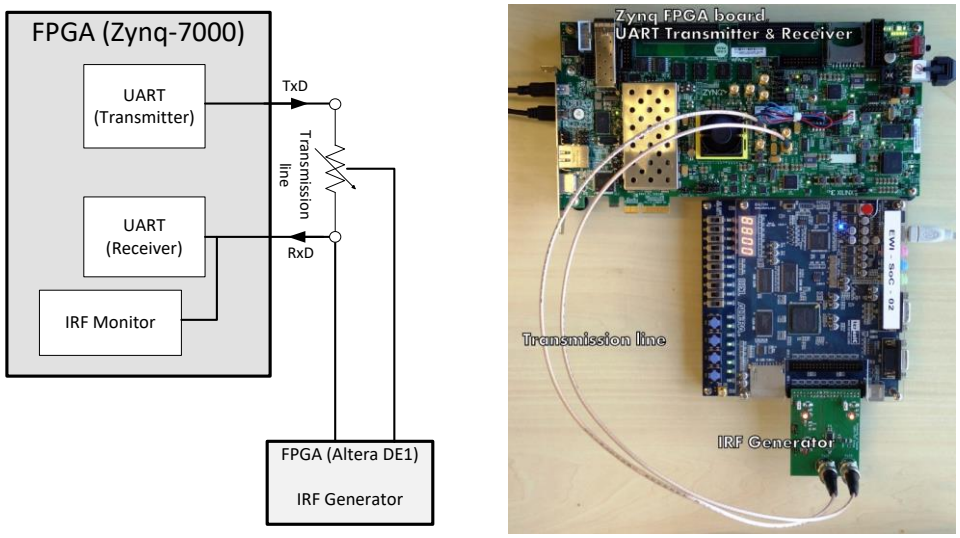
Figure 9. The communication window on a PC (USB-based) to set the IRF generator parameters. Compare this to the simulation model in D1.1.

The FPGA is responsible to synchronize and generate the control signals for the on-board programmable switches and potentiometers on the PCB board. The desired resistance range is provided by a combination of fast digital potentiometers and fixed range resistors. The IRF generator has been explained in deep in deliverable D1.2. Also a software equivalent for the IRF generator for IRF simulation purposes in Cadence is available, and has a similar GUI as tits hardware equivalent.

2.11 Intermittent Resistive Fault monitor

Related WP: WP3 and WP4
Partners: UT

In order to demonstrate IRF detection at board level a Universal Asynchronous Receiver Transmitter (UART) has been used as an example. A UART transmitter and a receiver were designed at logic gate level and implemented on the FPGA on Zynq-board. IRFs can be injected in the transmission line between the transmitter and the receiver by our hardware IRF generator (see Figure 10).



a) schematic b) board setup

Figure 10. Demonstration set-up of the IRF generator and IRF monitor a) schematic b) board setup

The IJTAG 1687 standard has been used for fault management and fault localization. Our IRF monitor has been wrapped by an IJTAG wrapper, and an IJTAG network has been developed and implemented on the FPGA of the Zynq ZC706 Evaluation Kit (see Section 3) to transmit information from the IRF monitor. When the IRF monitor detects an IRF, it raises a *warning* signal. This signal and the amplitude of violation will be transmitted through the IJTAG network to a PC and will be shown on a screen. This platform is shown in Figure 11.

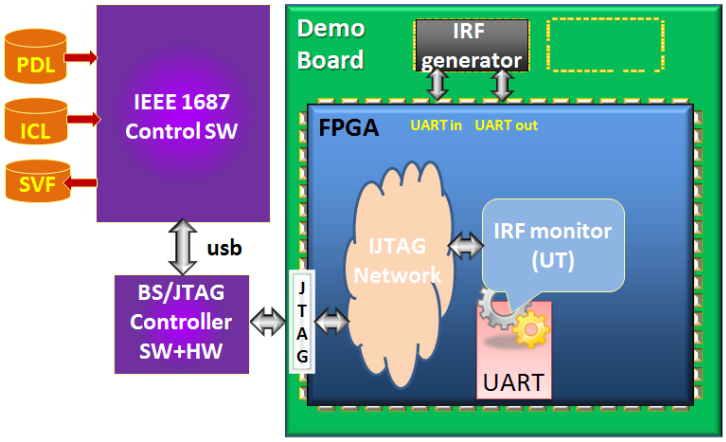


Figure 11. IRF monitor demonstration platform

2.12 QuadDPMO – Understand defect occurrences

Related WP: WP1
Partners: ASTER

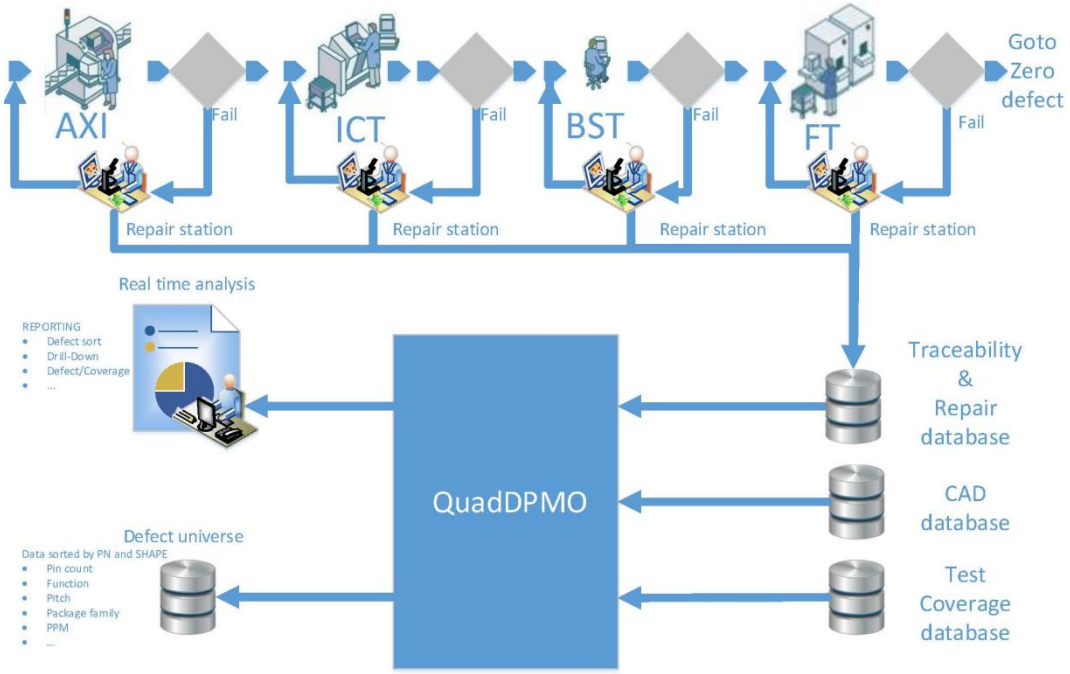


Figure 12. QuadDPMO synopsis.

Within the scope of the BASTION research program, a new concept is being developed by ASTER, to analyze the NFF phenomenon using a software tool. The QuadDPMO algorithm application is under development to:

- Collect & understand, in-real time, the DPMO,
- Group defect labels and root causes, by defect class,
- Compute long term, medium term and short term DPMO metrics,
- Investigate common areas of occurrence for each defect class.

QuadDPMO produces many reports where test data can be analyzed by site, period of time, test station, batch ID and product ID. It gives access to detailed reports where the defect family, defect code or defect label, can be sorted by various attributes, such as: pin count, pitch, mounting technology, mounting side, JEDEC shape, manufacturer, component function, board complexity, board type, etc.

The chart (pie or bar graph) color code is automatically transferred to the layout, or schematic views, in order to verify that the defect occurrence is linked with a physical or logical location (functional block).

ASTER has designed an innovative test coverage matrix, to classify defect occurrences against test coverage. A typical managerial report is created, where in one page, all critical information is presented: “Convert data to information” since information is used to make decisions.

There are 4 cases detailed in Figure 13:

- Faults which have been detected and are subject to be covered: This number shows the effective test coverage which is used. The difference between this number and total test coverage highlights the test coverage which has been produced but never used due to “no defect” occurrence. It opens the opportunity for cost savings with a test development driven by defects that really occur.
- Faults which have not been detected and are not covered. It is the typical definition of Insufficient Coverage.
- Faults which are detected in contradiction with the coverage: the test coverage is clearly under estimated. The coverage report generated by the test equipment must be updated to reflect the true test efficiency.
- Faults are not detected and are supposed to be covered: the test coverage is over estimated. It contributes to increase escape rates and NFF.

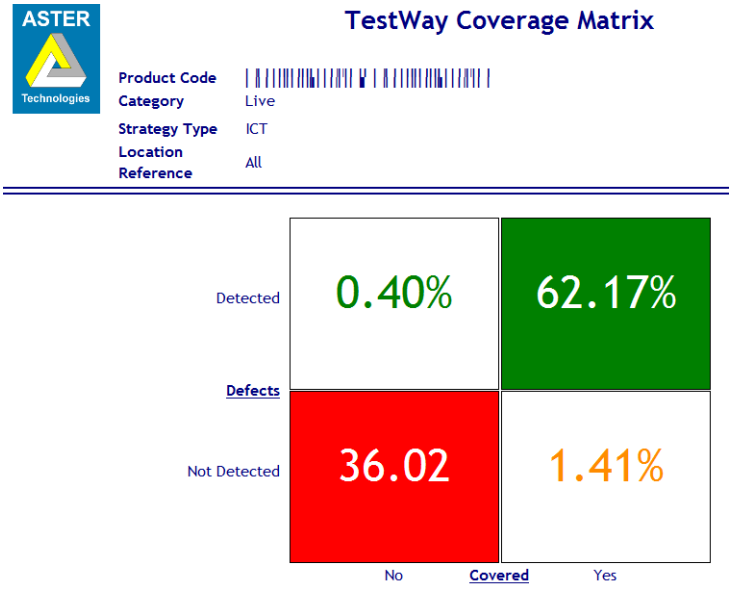


Figure 13. Innovative TestWay Coverage Matrix

2.13 Test Coverage Analysis in TestWay Express

Related WP: WP1
Partners: ASTER, TL, TUT, PDT

The test coverage tool from ASTER has been updated to support new classes of board-level faults as defined in D1.1. It includes: Aging defects, Intermittent Resistive Faults and At-Speed defects as shown in Figure 14 (see BASTION test phase).

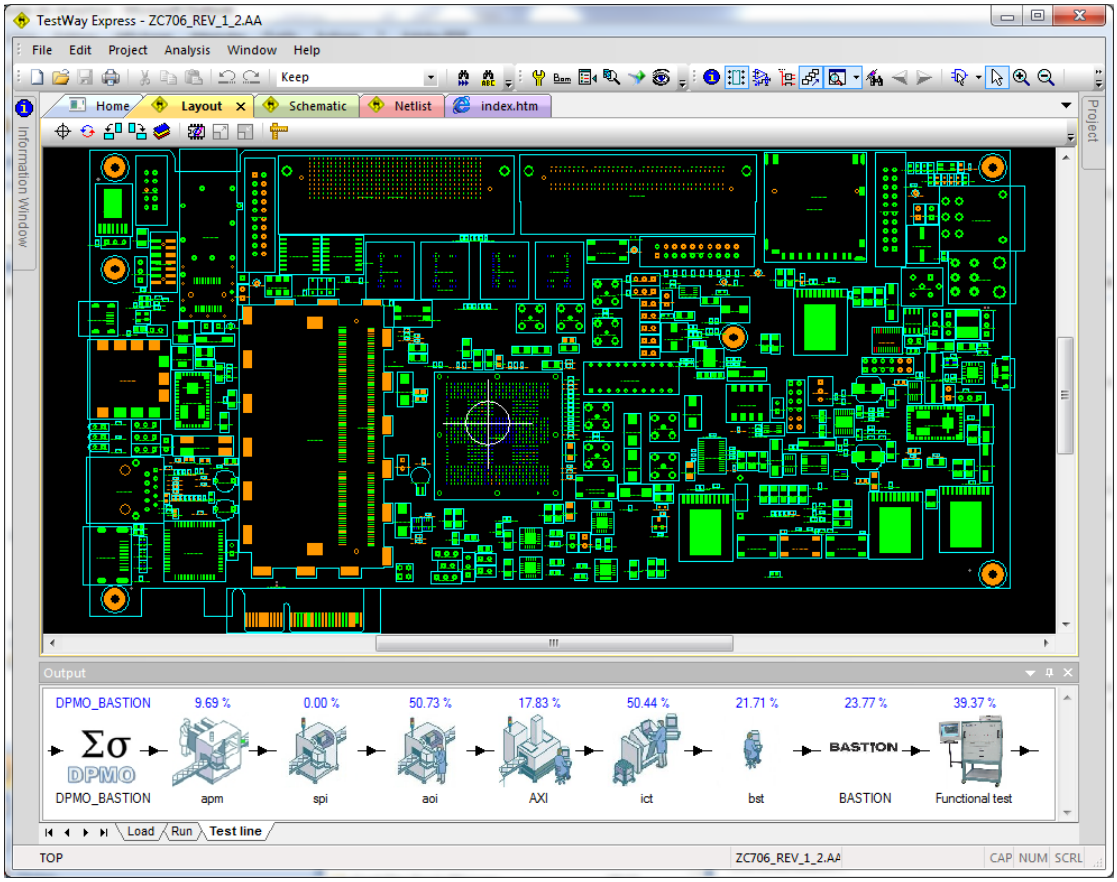


Figure 14. Graphical test coverage representation for the BASTION demo platform

As the result, this tool is used in the BASTION demonstration set up to estimate the test escape rate improvement achieved after inclusion of the novel BASTION methods and tools into the board test strategy. Figure 14 graphically shows test coverage for the Zynq ZC706 Evaluation Kit (see Section 3), while it is also possible to evaluate detailed fault tables and compare the BASTION-driven fault coverage improvement against the original typical state-of-the-art test coverage. This functionality was used to demonstrate the progress on KPI5 (see Section 3 for details).

In addition, a DPM table is produced by QuadDPMO which highlights the amount of PPM per defect category. The real extracted PPM information can be imported by the test coverage tools in order to measure the deviation between the theoretical defect rates (used in Figure 15) and the real defects, extracted from the traceability & repair database rates (used in Figure 16).

Figure 15. Test Coverage with standard DPM

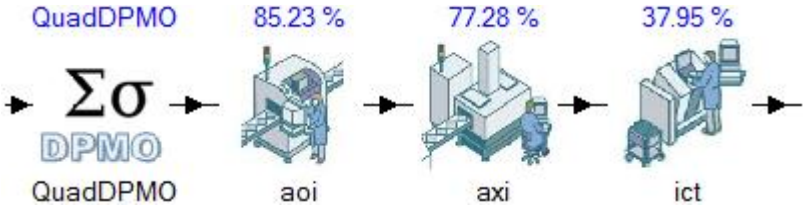


Figure 16. Test Coverage with real DPM

There are two strategic benefits for the end customer/user:

1. If the theoretical defect rate is higher than the real DPM, test coverage has been developed to detect defects that do not occur, or occur rarely. This provides an opportunity to optimize the test strategy, by focusing only on the most probable defects. This optimization could be static (done one time for the overall board production), or dynamic (adjusted in real time by disabling some tests when the DPM is decreasing). The dynamic mode can facilitate an increased throughput, when a test becomes a bottleneck of the production line. Alternatively, specific test lines could be selected based on their capability to detect certain defects, thereby tuning the test line to the product being manufactured. This capability could be expanded to allow OEM companies to select an EMS company, based on their process's DPM figures.
2. If the real DPM is higher than the theoretical defect rate, it means better test coverage is required on this component, in order to prevent a higher escape rate, which, in turn, contributes to the NFF phenomenon.

3 The BASTION Common Demonstration Platform

The main target of the demonstration work package is twofold: to demonstrate key achievements and to allow combined usage of tools and complementary techniques. Therefore a due effort has been spent by the BASTION consortium on defining a common vision towards the demonstration platform and studying technical feasibility of this target, which eventually has been successfully achieved.

3.1 HW Modules of the Demonstrator

Figure 17 shows a photograph of the demonstration hardware, which consists of the following modules:

- Xilinx Zynq-7000 All Programmable SoC ZC706 Development Board [8];
- RELY daughter board hosting RELY IC [6] with aging monitor;
- FMC adapter board (green) hosting RELY daughter board and JTAG loopback for Board BIST;
- FMC adapter board (brown) hosting extra JTAG devices for the BBIST demo;
- IRF generator board plus adapter shown in Figure 8 in Section 2.10, where Figure 9b shows also the complete setting (not shown in Figure 17 below).

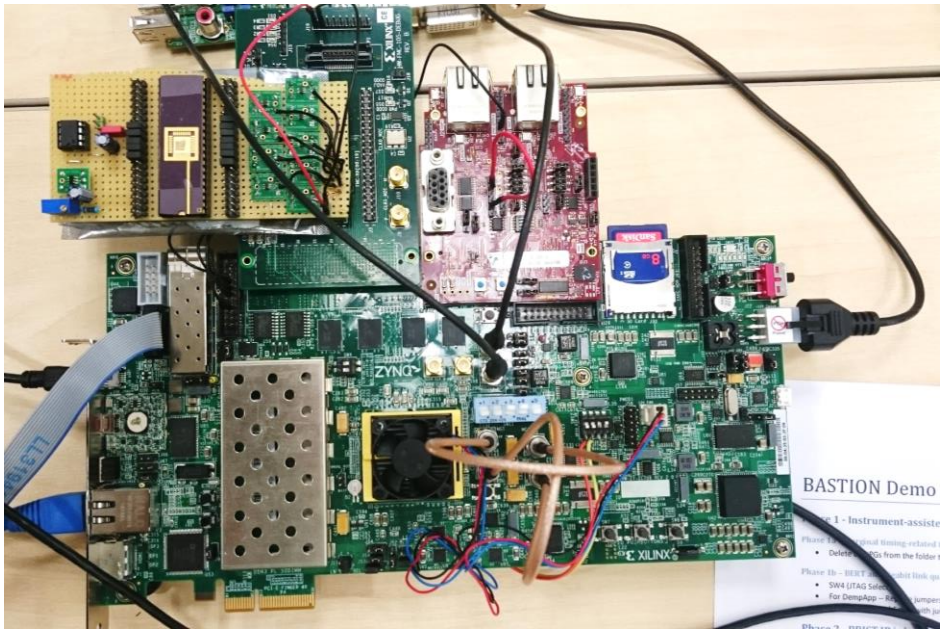


Figure 17. Test strategy (flow), BASTION contributions and respective demonstration phases

ZC706 development board plays the role of the main carrier hosting all other HW modules. The reasons for selecting this particular board have been explained in D6.1. The key aspects are:

- Double-core ARM Cortex A9 processor capable of Linux OS and Fault Management functions;
- FPGA logic large enough to accommodate
  - four processor cores and 4xNoC routers;
  - IEEE 1687 IJTAG network;
  - monitors, checkers, instruments;
  - Board BIST.
- DDR3 memory module and gigabit channels for marginal defect test demo;

- FMC connectors and UART for extension cards;
- Switches and LEDs used for fault injection and status indication;
- Availability of CAD data, including the full netlist used for fault modeling, test coverage calculation and test escape rate estimation in TestWay Express.

The ZC706 board represents an example of a modern complex high-performance system under test, suitable for such applications as e.g. high-definition video streaming for HDMI-based display applications and for development of networking applications with Gigabit Ethernet. For more details, see deliverable D6.1.

3.2 Overall Structure of the Demonstration Platform

Figure 18 shows the overall composition of the common demonstration platform and its main components detailed in D6.1. Software tools, their interaction, and data interfaces are shown on the left hand side, while the right hand side presents the HW and firmware implemented inside the FPGA. These components are described in Section 2 of this deliverable.

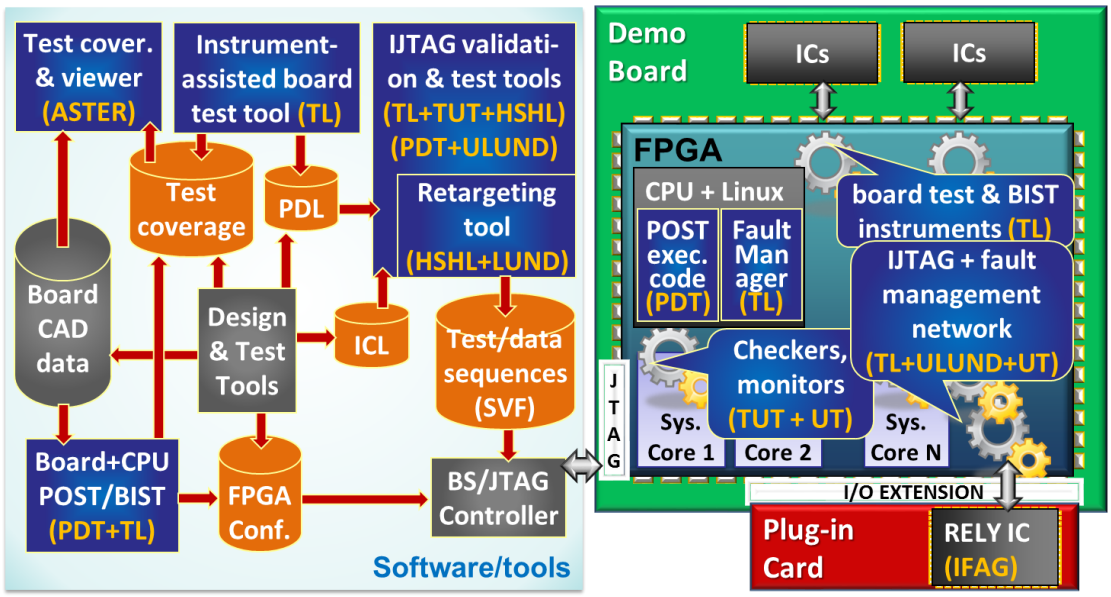


Figure 18: Structure of the Common Demonstration Platform

3.3 FPGA Configuration (firmware)

The firmware synthesized onto the FPGA has become rather complex. It includes:

- four processor cores and 4xNoC routers with checkers (see Figure 19);
- re-configurable IJTAG network (see Figure 20) for various experiments including fault injection, test and validation of the IJTAG itself as well as fault management;
- embedded instrumentation, monitors and checkers for implementing fault-management and on-line test scenarios incl. SBST/POST, IRF monitoring, etc.;
- connection with microprocessor for instrument-assisted functional test scenarios (SBST);
- high-performance instruments for advanced board test for TRFs and marginal defects as well as for implementing Board BIST/POST scenarios.

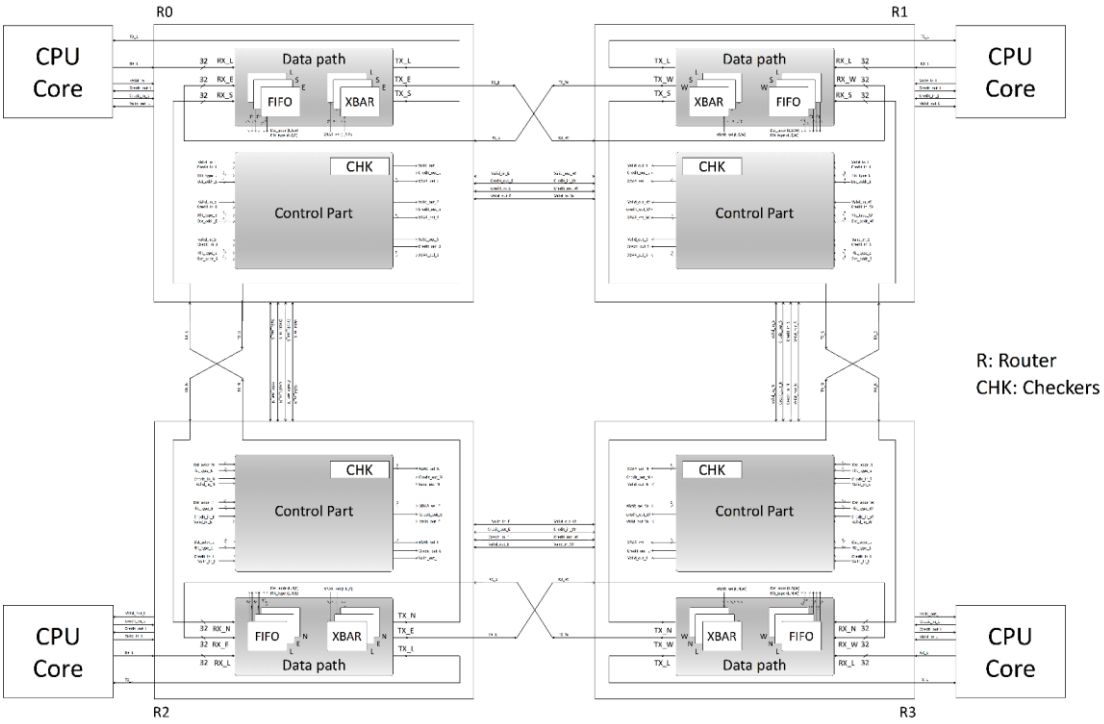


Figure 19: System Under Monitoring in fault management scenario (inside FPGA)

Figure 20 shows TreeBalanced IJTAG network benchmark from the ITC'2016 set [15] that was extended to support the asynchronous emergency signaling network (based on FCX registers) and mapped to the system under monitoring by connecting NoC data integrity checkers and RELY aging monitor to suitable test data registers.

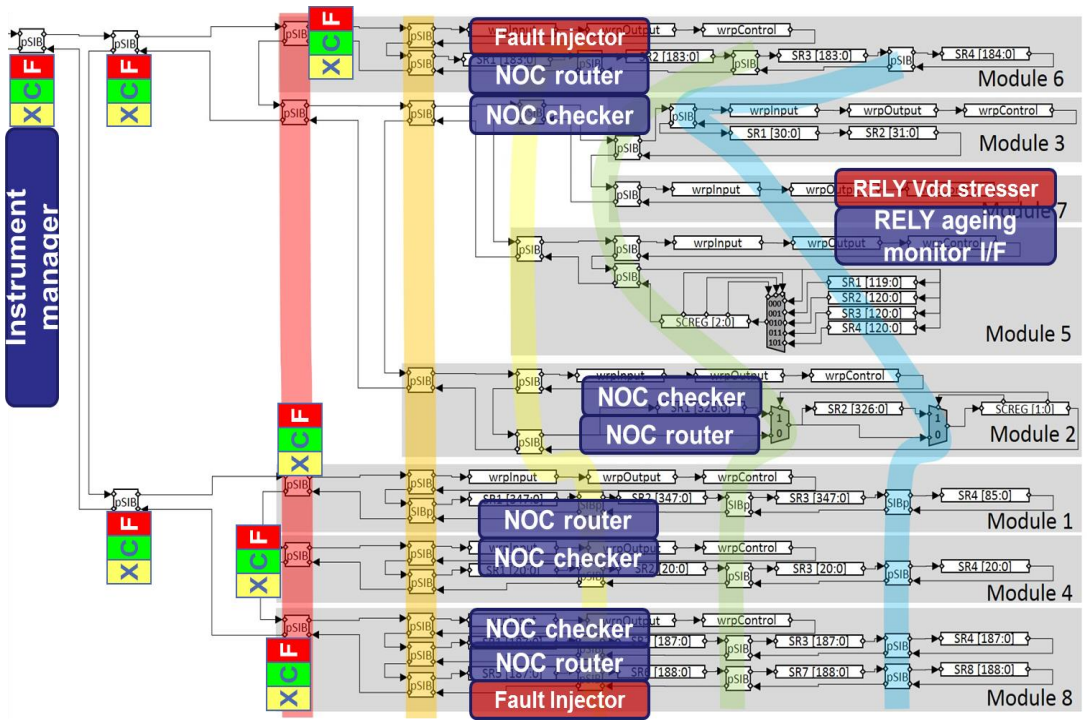


Figure 20: ITC'2016 IJTAG Network Benchmark Mapped to the System Under Monitoring

In addition to the firmware the FPGA device is hosting Linux operating system with fault management software running on CPU cores.

3.4 Demonstration Storyline

A selection of BASTION contributions related to the demonstrator is shown on the upper part of Figure 21, while the bottom part presents the demonstration flow. Both parts are mapped on the system test planning, test development, test application and in-field monitoring flow (a simplified version).

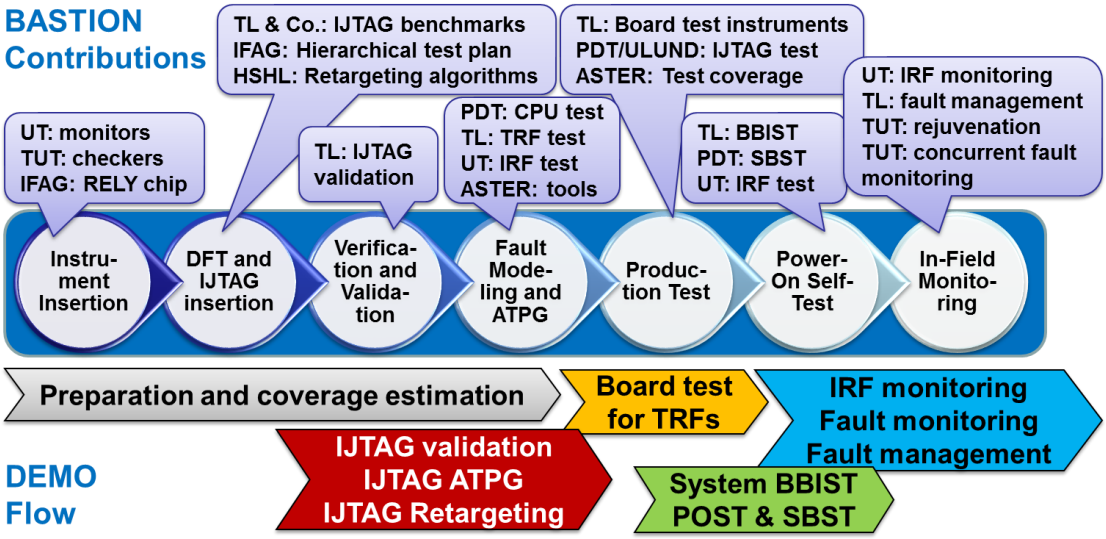


Figure 21. Test strategy (flow), BASTION contributions and respective demonstration phases

This flow has become an outline of the BASTION demonstration scenario that consists of the following main phases:

1. Instrument-assisted board test for TRFs and marginal defects;
2. Board BIST IP is loaded and the board is self-tested;
3. SBST IP is loaded and CPU is self-tested;
4. IRF fault injection and monitoring demonstration;
5. Monitoring IP including IEEE1687 infrastructure is loaded for:
   - IEEE1687 infrastructure validation and test demo;
   - Optimized retargeting demonstration on TreeBalanced network;
   - Fault management demonstration including checkers and RELY chip;
6. Test coverage comparison and evaluation and test escape rate calculation.

3.5 Achieving Progress on KPI5

The joint effort of all partners in bringing up the demonstrator that required connecting all respective tools, exchanging description formats, and reporting the fault coverage resulted in our ability to address **KPI5: Reducing test escapes and impact of NFF by, at least a factor of 2**.

All individual coverage reports from BASTION tools have been collected and imported by TestWay Express (see Section 2.13), which in its turn was extended to support new coverage metrics proposed by BASTION (WP1) such as TRFs, IRFs and marginal defects as well as specific types of faults and wear-out effects targeted by checkers and

monitors. As a result, the extended fault universe required the combined usage of both the traditional board test methods (not capable of covering the new fault types) and the new BASTION test methods (having a limited coverage of traditional fault types).

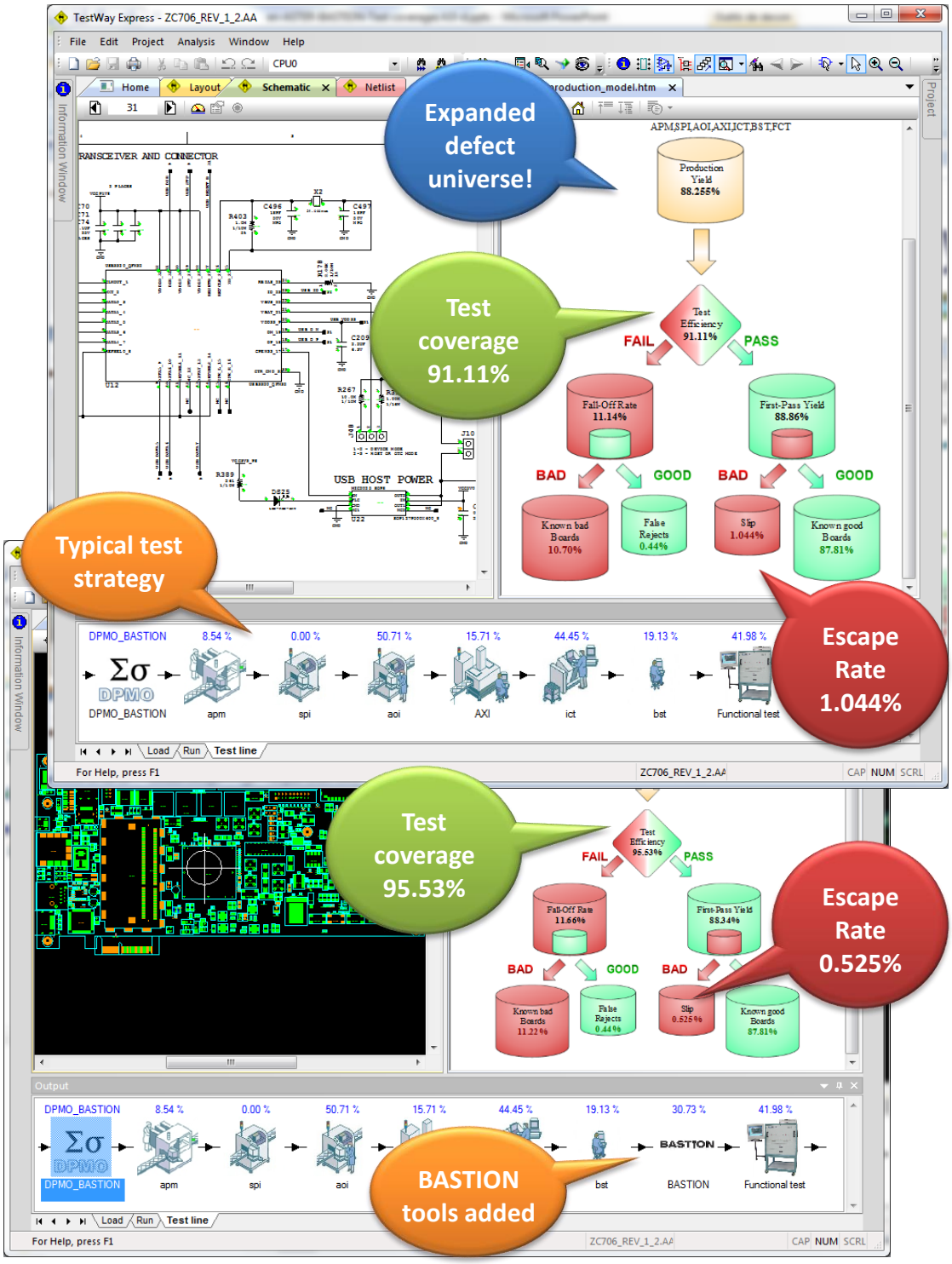


Figure 22: Test strategies with and without BASTION tools: coverage and escape rate

Based on incoming DPM values and the actual test coverage, TestWay can calculate estimated resulting test escape rate. We have calculated test escape rates for two different scenarios:

1. Classical board test methods without BASTION tools produced test coverage of 91.11% of the extended defect universe. With a realistic incoming DPM, this resulted in expected test escape rate of 1.044%
2. BASTION tools together with classical methods improved the coverage up to 95.53%, which resulted in test escape rate of 0.525%

This experiment is illustrated in Figure 22, where the upper screenshot shows the first scenario and the other one corresponds to the second case. As the result, the test escape rate improvement reached the value of 1.99, which is very close to the initial target factor of 2. However, we must admit that it is just a calculated estimation. The reality may be either better or worth depending on the real DPM related to marginal defects or TRFs. If their impact is less than expected, then the gain might be smaller, or otherwise – bigger. Getting real numbers requires, however, long learning loop that involves data from actual production where the new BASTION technologies would have been implemented and their real impact measured.

4 Conclusions

BASTION project has resulted in an impressive number of prototype tools. We have presented 13 individual items whereas 8 of them were developed by industrial partners or in a direct collaboration with them. Many tools are prototypes at high technology readiness level, which promises a successful exploitation and commercialization of BASTION project results.

A tremendous effort has been spent by the BASTION consortium on defining a common vision towards the demonstration platform as well as on actual development work, which eventually has led to a successful outcome.

The ZC706 board that was selected as the main HW carrier represents an example of a modern complex high-performance system under test, suitable for such applications as e.g. high-definition video streaming for HDMI-based display applications and for development of networking applications with Gigabit Ethernet. The Xilinx Zynq-7000 All Programmable SoC ZC706 Evaluation Kit is distributed together with CAD data, including netlist, which made it possible to calculate fault coverage and perform testability analysis.

The main experimental result of the joint collaboration of all BASTION partners in frames of WP6 was the ability to demonstrate our progress on **KPI5: Reducing test escapes and impact of NFF by, at least a factor of 2**. This was further facilitated by the necessity of bringing up the demonstrator that required connecting all respective tools, exchanging description formats, and reporting the fault coverage from individual tools. All these coverage reports from BASTION tools have been collected and imported by TestWay Express (see Section 2.13), which in its turn was extended to support new coverage metrics proposed by BASTION (WP1) such as TRFs, IRFs and marginal defects as well as specific types of faults and wear-out effects targeted by checkers and monitors.

As the result, the estimated test escape rate improvement reached the value of 1.99, which is very close to the initial target factor of 2. However, we must admit that it is just a calculated estimation. The reality may be either better or worth depending on the real DPM related to marginal defects or TRFs. Getting real numbers requires, however, long learning loop that involves data from actual production where the new BASTION technologies would have been implemented and their real impact measured.

5 References

[1] *Big Trouble with "No Trouble Found" Returns,* Accenture Report, 2008, Available at: http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture_Returns_Repairs.pdf

[2] *Embedded Instrumentation: Its Importance and Adoption in the Test & Measurement Marketplace: A Frost & Sullivan White Paper*, Frost & Sullivan, 2009. Available at: www.frost.com/prod/servlet/cio/201638885

[3] Pietro Saltarelli, Behrad Niazmand, Ranganathan Hariharapn, Jaan Raik, Gert Jervan, Thomas Hollstein. Automated Minimization of Concurrent Online Checkers for Networks-on-Chip, *Proceedings of the 10th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC 2015)*, Bremen, 2015.

[4] E. Sanchez; M. Schillaci; G. Squillero, *Evolutionary Optimization: the µGP toolkit*, Springer, ISBN: 978-0-387-09426-7, 2011

[5] zamiaCAD framework web page, [http://zamiaCAD.sf.net]

[6] RELY - Design for RELIABILITY of SoCs for Applications like Transportation, Medical, and Industrial Automation, http://www.rely-project.eu/

[7] N. Pour Aryan, G. Georgakos, D. Schmitt-Landsiedel, “Reliability Monitoring of Digital Circuits by in situ Timing Measurement ” in 23rd International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), 2013.

[8] Xilinx Zynq-7000 All Programmable SoC ZC706 Evaluation Kit [http://www.xilinx.com/products/boards-and-kits/ek-z7-zc706-g.html]

[9] R. Cantoro, M. Montazeri, M. S. Reorda, F. G. Zadegan and E. Larsson, "On the testability of IEEE 1687 networks," 2015 IEEE 24th Asian Test Symposium (ATS), Mumbai, 2015, pp. 211-216.

[10] R. Cantoro, M. Palena, P. Pasini and M. S. Reorda, "Test Time Minimization in Reconfigurable Scan Networks," 2016 IEEE 25th Asian Test Symposium (ATS), Hiroshima, 2016, pp. 119-124.

[11] R. Krenz-Baath, F. Ghani Zadegan, and E. Larsson, "Access Time Minimization in IEEE 1687 Networks," in *Proc. International Test Conference (ITC)*, 2015.

[12] F. Ghani Zadegan, R. Krenz-Baath, and E. Larsson, "Upper-Bound Computation for Optimal Retargeting in IEEE 1687 Networks," in *Proc. International Test Conference (ITC)*, 2016.

[13] M. Jenihhin, G. Squillero, T. Copetti, V. Tihhomirov, S. Kostin, M. Gaudesi, F. Vargas, J. Raik, M. S. Reorda, L. B. Poehls, R. Ubar and G. Medeiros, "Identification and Rejuvenation of NBTI-Critical Logic Paths in Nanoscale Circuits," *Journal of Electronic Testing-Theory and Applications (JETTA)*, vol. 32, no. 3, pp. 273-289, 2016.

[14] F. Pellerey, M. Jenihhin, G. Squillero, J. Raik, M. Sonza Reorda, V. Tihhomirov, R. Ubar, "Rejuvenation of NBTI-Impacted Processors Using Evolutionary Generation of Assembler Programs," *2016 IEEE 25th Asian Test Symposium (ATS)*, Hiroshima, 2016, pp. 304-309.

[15] A. Tsertov, A. Jutman, S. Devadze, M. Sonza-Reorda, E. Larsson, F. Ghani Zadegan, R. Cantoro, M. Montazeri, R. Krenz-Baath, “A Suite of IEEE 1687 Benchmark Networks” in *Proc of 47th IEEE International Test Conference (ITC’2016)*, Fort Worth, TX, USA, Nov 15-17, 2016, paper 6.1.