DEGREE PROJECT, IN , SECOND LEVEL
STOCKHOLM, *SWEDEN 2014*

HW Fault Coverage Analysis

DIMITRIOS BARDIS

KTH ROYAL INSTITUTE OF TECHNOLOGY

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

HW Fault Coverage Analysis

Examiner/Supervisor: Elena Dubrova(KTH)
Supervisor: Gunnar Carlsson (Ericsson)

DIMITRIOS BARDIS

KTH Industrial Engineering
and Management

Master of Science Thesis
Stockholm, Sweden 2014

TRITA-ICT-EX-2014: 42

Acknowledgements

I would like to thanks Ericsson AB and in particular my manager Åke Strömberg and my supervisor Gunnar Carlsson for their support, ideas, guidance and for believing in me for this project. I would also like to thanks Kim Petersén for his contribution to the Function Testing side of the Project.

Moreover I would like to notice the great contribution of Pete Collins (Business Development Manager at ASTER Technologies Ltd), Christophe Lotz (President of ASTER Technologies Ltd) and their colleagues in ASTER Technologies Ltd. They helped me with their technical support and technical training on their DfT Tool. Regarding Aster's proprietary information in this project, it should be mentioned that they are used by courtesy of Aster Technologies and the corresponding information are considered as Intellectual properties of Aster Technologies.

In the end I would also like to thank my supervisor Elena Dubrova for believing in me and my parents for their endless support in all levels.

Table of Contents

Abstract

In Ericsson Radio Base Station (RBS) products a very high quality is crucial. To achieve such a high quality, the production test must be capable of detecting all potential faults introduced in the production process.

During the production phase it is very important to achieve the maximum coverage possible on a HW implementation. The major test strategies that will be evaluated in this Project will be BSCAN (Boundary Scan Testing), FT (Functional Testing) and AOI (Automated Optical Inspection) and the PCB that will be tested under these test strategies is TCU board.

Searching the commerce for a valuable Fault Coverage Analysis tool is the basic step in order to test the PCB. Next, a suitable method for the use of the tool will be reported to Ericsson and recommendations also to Ericsson AB on whether to use the tool or not should be the conclusion of this Project.

Table of Figures

List of Tables

Glossary

AOI	**Automated Optical Inspection**
BSCAN	**Boundary SCAN**
FT	**Functional Testing**
BST	**BSCAN Testing**
AXI	**Automated X-RAY Inspection**
JTAG	**Joint Test Action Group**
CAD	**Computer Aided Design**
PCB	**Printed Circuit Board**
TRST	**Test Reset**
TCK	**Test Clock**
TMS	**Test Mode State**
TDI	**Test Data In**
TDO	**Test Data Out**
BOM	**Bill Of Materials**
DPMO	**Defects Per Million Opportunities**
ICT	**In Circuit Testing**
TWL	**TestWay Netlist**
BSDL	**Boundary Scan Description Language**

1 Introduction

In Ericsson's Radio Base Station (RBS) products, a very high quality is crucial. To achieve such a high quality, the production test must be capable of detecting all potential faults introduced in the production process

1.1 Background

Fault coverage is a very indicative feature when it comes to testing a PCB. It is very important to deliver to the customer a fault free product and that is why that product shall be tested as much as possible during the production phase.

1.2 Problem

The main problem is that today Ericsson has no formal means to analyze the fault coverage of the different types of tests (BSCAN, FT and AOI) during the production phase of the product.

1.3 Purpose

The purpose of this Master Thesis is to select a promising tool in order to test a quite complex system in the production phase for BSCAN, FT and AOI test strategies.

1.4 Goal, Benefits, Ethics

The goal of this Master Thesis is to propose a suitable method for the use of the tool that is selected for detecting the maximum number of faults in a HW implementation. Ericsson will get the findings and reports on how to use the tool and also recommendations on whether it is efficient to use this tool or not.

• Study previous work on methods and related tools for fault coverage analysis, and report state of the art.

• Investigate the market to find out what commercial tools exist for fault coverage analysis.

• Choose a promising candidate tool, and evaluate it for the different test scenarios of production test.

• Report findings and propose a suitable method for the use of the tool.

• Conclusions and recommendations on whether to use the method and tool at Ericsson.

1.5 Methodology

The project will use the quantitative method to draw conclusions. Quantitative method is used in order to represent the fault coverage analysis results and make a decision on whether it is worth of using the selected tool or not.

1.6 Delimitations

The study considers all the possible aspects that could affect the effectiveness of the selected tool. However, it should be mentioned that the selected tool supports more attributes than the PPVS attributes that are analysed in this project and that means that the user is not limited to 4 attributes only. (Presence, Polarity, Value and Solder). Lastly, in field test and repair test are not considered in this project.

1.7 Outline

Chapter 2 contains all the necessary information that the reader needs to know in order to be able to understand the basic concepts of the main production test strategies.

Chapter 3 contains all the major commercial tools and the basic functionality information for each one of them.

Chapter 4 contains a short comparative analysis between all the commercial tools, from previous chapter, in order to decide which one to select.

Chapter 5 contains a suitable method for the use of the tool regarding three test strategies that Ericsson AB is mostly interested in. These are BSCAN, FT and AOI.

Chapter 6 contains a discussion on various issues regarding advantages of the tool and things for reconsideration regarding functionality of the tool and possible further work.

2 Basic Background

In printed circuit boards there can be cases when a specific part is malfunctioning or is not working at all. For example a digital output pin may not toggle correctly. In this case the digital output pin is considered to be defective and this case is called fault.

In general a defect can be considered as a deviation from a normal situation but it cannot always be detected as a fault. For example insufficient solder is a defect, since it is a deviation from a normal situation (normal would be sufficient soldering) but it may not be detected as a fault if it is not letting the system malfunction.

Fault coverage (FC) is the main topic that this project is based on. Fault coverage is basically the conditional probability that, given the existence of a fault, the test detects it.

$$FC = P(\text{fault detection} | \text{fault existence}) \quad [1]$$

Fault coverage is not always very representative of the actual fault coverage of the system. For example 90% fault coverage does not necessarily mean that 90% of all the potential faults were detected by the test program. The reason is that there can be more defects that may not be modeled by the test program and as a result they are not included in to this fault coverage figure.

At this point it is necessary to define a realistic list of all potential defects for a certain PCB (Printed Circuit Board). The list with the defects can be created based on the structure of the Board, the bill of materials (BOM) and the placement of the various modules on the board. The list in the end is called Defect Universe.

Before analyzing the defect universe it would be wise to make a short introduction to the DPMO definition. DPMO (defects per million opportunities) is basically process performance indicator and it describes the number of faults out of the total number of opportunities in a set of data. [23]

In order to calculate DPMO we need to know DPO (Defects per Opportunity) in the first place.

$$DPO = \frac{Number\ of\ Defects}{(Number\ of\ Units)x(Opportunities\ for\ a\ defect\ per\ unit)}$$

In our Project a unit is considered to be a soldering unit or another unit that is a point for testing. Since we have already calculated DPO we can calculate DPMO like below:

$$DPMO = DPO\ \ x\ 1000000$$

In order to consider all defects including design validation and testing, we need 3 main defect classes:

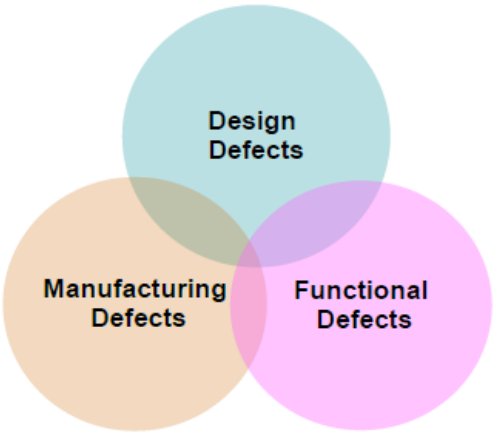


Figure 1. Defect Universe

Design Defects: Defects that need to be detected by a design rules checker or by electrical DfT.

Manufacturing Defects: Defects regarding component or interconnects during the production phase

Functional Defects: Defects that are affecting the function of the system.

3 Main Standards for modelling fault coverage have been developed. Those are [3]:
MPS: Philips Research
PPVS: ASTER Technologies
PCOLA-SOQ: Agilent Technologies

Table 1: MPS-PPPVS-PCOLA SOQ

MPS	PPVS	PCOLA-SOQ
Material	Value	Correct
		Live
Placement	Presence	Presence
		Alignment
	Polarity	Orientation
Solder	Solder	Short
		Open
		Quality

The most commonly used and widely spread defect standard is the PCOLA-SOQ developed by Agilent Technologies. Later FAM standard, regarding functional faults, was added to it by INEMI thereby forming the final PCOLA-SOQ-FAM standard.

FAM tests are basically verifying a Feature on the board at a specific speed taking a measurement.

Regarding PCOLA-SOQ-FAM[14]:

Table 2 : PCOLA SOQ FAM Table

Structural Devices	P	Presence
	C	Correctness
	O	Orientation
	L	Live
	A	Alignment
Structural Interconnects	S	Short
	O	Open
	Q	Quality
Functional Devices & Connections	F	Feature
	A	At-Speed
	M	Measurement

2.1 Structural Attributes

DEVICE LEVEL

Presence: This property confirms whether a component of the PCB is present or not. However, presence value cannot specify whether the component is the correct one or not. It just specifies whether there is a component in that place or not. [13]

Correctness: This property is meaningful only if the previous one (Presence) shows that the component is present. Correctness value demonstrates that the component that is already detected (by presence property) is the correct one. For instance, Correctness property can tell whether a resistor is a 10kohm or 1kohm resistor. [13]

Orientation: Orientation basically shows whether there is any rotation of the component during the placement or not. The rotation that can be noticed, in this case, shall be multiple of 90 degrees. For example Automatic Optical Inspection can specify the orientation by checking the polarity mark on the component. [13]

Live: This property ensures that a component is performing some function without confirming the exact functionality. [13]

Alignment: Alignment property gives us the confirmation that the component is correctly aligned and that also all the pins of this component are soldered to the board pads according to the specifications given. [13]

INTERCONNECTS LEVEL

Short: Short property shows whether two pins are shorted or not. This can happen due to excess soldering or bent pins. It can also happen that two pins, that belong to the same electrical net, may be shorted thereby making it difficult to get detected. [13]

Opens: In this case, we get information on whether a pin is connected correctly to the board pad or not. [13]

Quality: This property confirms whether the quality of the overall structural integrity of the system is sufficient. For example quality of solder is checked. [13]

2.2 Functional Attributes

Feature: This property confirms during functional testing whether a feature is present or absent. If the presence or absence of the feature can be determined during Function Testing the Feature property gets the value '1.00'. Otherwise it is assigned value '0.00'. [18]

At-Speed: At-Speed property shows whether the pin/interface/feature is tested at min/mid/max speeds or at subset of min/mid/max speeds. In the first case the value is '1.00' and in the second case the value is '0.500'. Value '0.00' corresponds to no-test case. [18]

Measurement: This property gets value '1' when the measurement confirms performance of the component (performance regarding voltage, current, etc.). If no measurement is taken, the property gets value '0.00'. [18]

Every technique detects some faults and some of them may be common between various techniques. However, there is no test strategy that can actually cover all the possible faults. For this reason, the combination strategy is used so that we achieve to cover the maximum number of faults that can be covered by combining different test strategies.

For example, when using the PCOLA-SOQ standard the following figure helps us understand the way that the different test strategies are combined in order to give us a better fault coverage. [4]

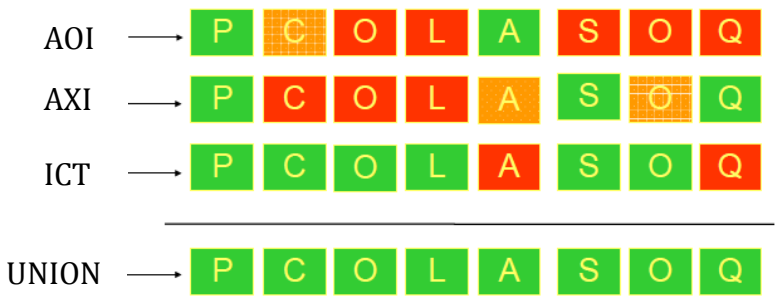


Figure 2. AOI-AXI-ICT Combination

We can also do the same for the PPVS standard where the coverage of each device is split into 4 coverage facets:

- Presence: The same terminology with Presence attribute from PCOLA-SOQ standard applies here.
- Polarity: The same terminology with Orientation attribute from PCOLA-SOQ standard applies here.
- Value: The Value attribute indicates if the detected part is the right one and whether it performs some function or not.
- Solder: This attribute indicates in which level the device has been tested for solder-related faults (such as bridging, open or short faults).

Table 3. PPVS -PCOLA SOQ

PPVS	PCOLA/SOQ
Value	Correct Live
Presence	Presence Alignment
Polarity	Orientation
Solder	Short Open Quality

The following figure shows a combination of test strategies that cover structural tests, visual inspection and functional tests. [3]

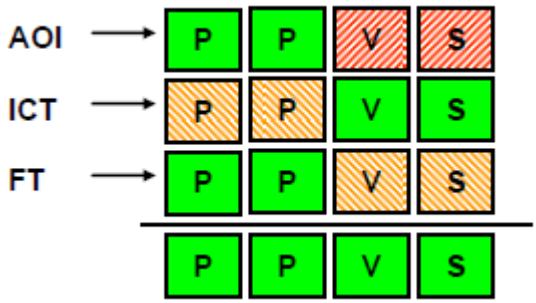


Figure 3. AOI-ICT-FT Combination

In both cases we notice that we achieve the maximum coverage by combining different test strategies.

2.3 AOI (Automatic Optical Inspection)

From reference [7] we can find information about AOI. AOI is a test method that is used during the production phase and it is based on inspecting the PCB device for various defects. The most common defects are solder defects, missing components, incorrectly placed components and open/short circuits.

In comparison to the manual operators that were performing before the PCB inspection, AOI is far more quick and accurate. AOI systems are basically

responsible for bare board inspection, solder paste inspection and component placement. In addition to this, pre-reflow and post-reflow condition of the components is checked.

It is very important to detect problems as soon as we can in the manufacturing phase since later it will cost a lot to fix them. For instance, if during the production phase there is a fault on a PCB that is not detected then all the rest PCBs will be produced with the same fault.

AOI is very good solution for high-volume assembles due to low cost and low programming efforts. It is quite often combined with BSCAN testing, ICT, AXI and Functional Testing. It shall be mentioned that sometimes smaller circuit board designs need more an AOI test method rather than ICT method.

2.3.1 Process

An AOI system can acquire millions of pixels per second. These pixels are used for optical inspection and precision measurement.

AOI system scans the surface of the board which is lit by several light sources. A scanner or a high definition camera also observes the board in order to capture all the areas on the board. The inspection algorithms and lighting algorithms that are used by AOI systems may vary depending on the manufacturer of the system.[5]

2.3.2 Advantages and Disadvantages

AOI test method has a lot of benefits from a cost-related point of view. It has, however, some drawbacks also. Those are mentioned below and are mainly coming from references [20], [22]. But before mentioning them it is good to know a few things about reflow phase[21].

Reflow soldering is basically a procedure that is happening during the reflow process that is responsible for attaching an electrical component to the contact pad by using a solder paste. The technique in this case is based on the fact that controlled heat will melt in the end the soldering and that will connect the electrical component with the contact pad permanently.

First of all it is much cheaper to detect PCB defects in the process monitoring phase rather than final test and inspection. The reason why is the fact that detection at the earliest point (and correction consequently) prevents producing more faulty PCBs and as a result saves a lot of cost.

Moreover, problems like missing components or misplaced components are controlled during the component placement inspection which is before reflow phase. That lowers a lot the post-reflow effort and cost.

Another important feature of AOI that makes it preferable in comparison to X-ray inspection is the fact that post-reflow cost is less than that of X-ray

inspection for detecting defects such as bridges, broken joints and other kind of solder defects.

One of the most important advantages of AOI test method is the flexibility that we can have and the short test program development time. There are many technical parts that were updated in order to acquire a better performance. New ways of lighting, increased magnification and more sophisticated algorithms contribute to higher test speed and better performance regarding the speed of fault detection.

However, AOI has a few drawbacks that should be stated at this point. In the first place, it is very reasonable to state that only the visible defects can be detected. As a result the coverage in this case would not correspond to the real one since there would be more defects that would not be included in that coverage.

Another drawback of the AOI test method is that in some cases a false alarm can be activated and that makes the test method not so trustable. A false alarm can be triggered when there may be for example a misinterpretation of oxidation as substrate.

AOI for a PCB board with components may inspect the following features:

- Component polarity
- Component presence/absence
- Excessive/ Insufficient solder joints
- Flipped component
- Height Defects
- Insufficient paste around Leads
- Solder bridges
- Wrong part

2.4 BSCAN

2.4.1 BSCAN Functionality

Boundary Scan is a test strategy that is used for testing interconnects on PCBs. The Joint Test Action Group (JTAG) developed a specification for boundary scan testing that was standardized in 1990 as the IEEE Std. 1149.1-1990. In 1994, a new language was added to the JTAG research which describes the logic structure of the IEEE Std 1149.1 compliant devices. The name of this language is BSDL (Boundary Scan Description Language). Today Boundary scan is now mostly synonymous with JTAG. [9], [14]

From ref. [8] we find basic information for BSCAN. BSCAN test strategy enables the user to test interconnects on the PCB without using physical test probes. BSCAN structure includes a multiplexer and latches to each pin on the device. BSCAN cells (test cells) have a two-way functionality. They can either capture data from pin or core logic signal, or force data onto pins. The data that is captured is shifted out and compared to the expected result. On the other hand, regarding the second functionality, the forced data are shifted into

the BSCAN cells. The path that controls all this procedure is called scan path or scan chain and it is shown in the figure below.

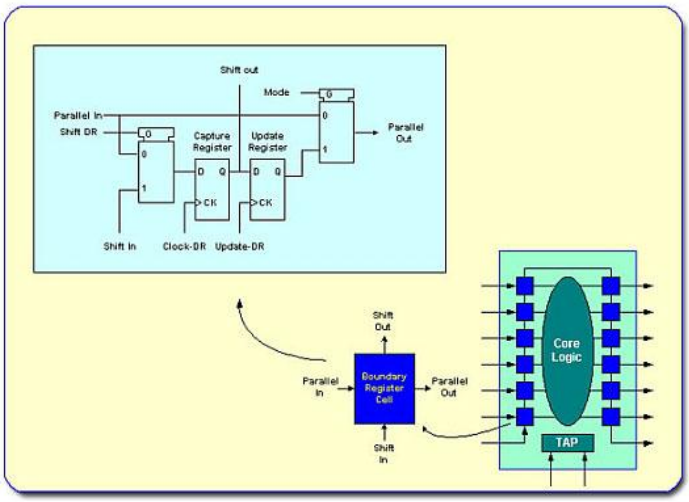


Figure 4. BSCAN Functionality

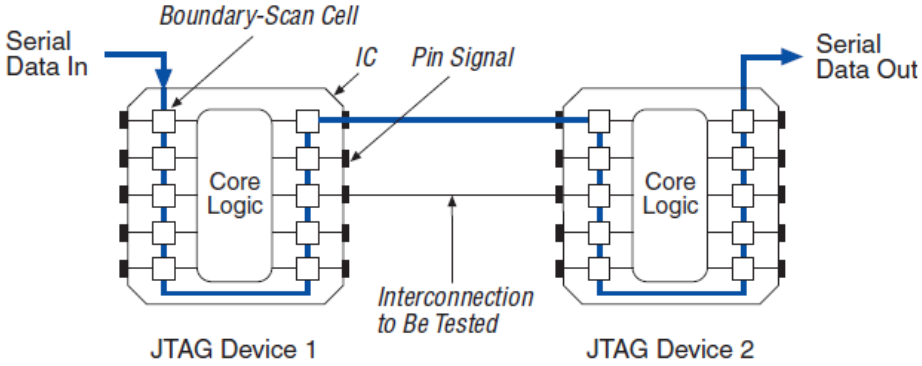


Figure 5. BSCAN Chain [Ref. 10]

Every test cell is programmed, via the scan chain, in order to drive a signal onto a pin. Then this signal is propagated across a distinct trace on the board. At the end of its destination, the signal meets a cell which we read in order to verify that the board trace connects properly the two pins. For example, if there is a short or an open defect at the trace, the correct signal value is not present at the destination pin and then we confirm that there is a fault.

The functionality of interconnect testing using BSCAN test strategy is depicted in the following figure. We can see two BSCAN compliant devices, U1 and U2, which are connected with four nets. U1 has four outputs that are driving the four inputs of U2 with some values. In this example, we assume that the circuit includes two faults: a short between Nets 2 and 3, and an open on Net 4. [8]

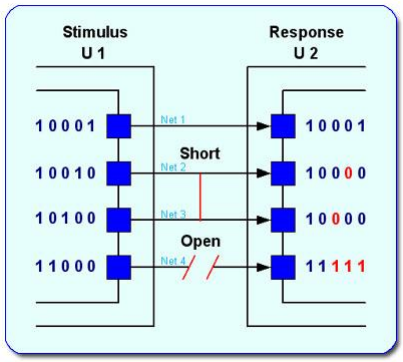


Figure 6. Interconnect Testing

In addition to this, in terms of logic, a short is considered to behave as a wired-AND and an open is sensed as logic 1. Then the tester is shifting the patterns that we can see in the figure from U1 to U2 so that we can check for any possible defect. The input values of U2 BSCAN register are then shifted out and compared to the expected results. For this specific example, based on the patterns that we chose in combination with the defects shown, we can notice that the results (marked in red) on Nets 2,3 and 4 are not the same with the expected values and in this way the tester detects the faults on Nets 2,3 and 4.

A more detailed presentation of the JTAG- device is shown below [11], [24]:

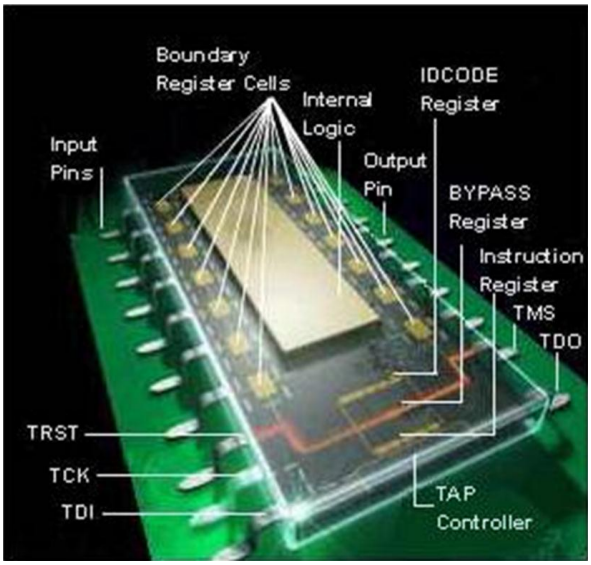


Figure 8.JTAG-Device Pins and Registers

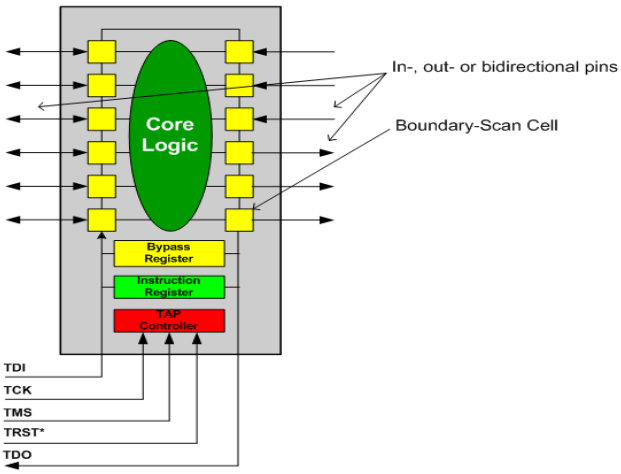


Figure 7. BSCAN Cell Pins

Table 4. JTAG TAP (Test Access Points) Interface Signals [Ref. 8]

Abbreviation	Signal	Description
TCK	Test Clock	Synchronizes the internal state machine operations
TMS	Test_Mode State	Sampled at the rising edge of TCK to determine the next state
TDI	Test Data In	Represents the data shifted into the device's test or programming logic. It is sampled at the rising edge of TCK when the internal state machine is in the correct state.
TDO	Test_Data Out	Represents the data shifted out of the device's test or programming logic and is valid on the falling edge of TCK when the internal state machine is in the correct state
TRST	Test Reset	An optional pin which, when available, can reset the TAP controller's state machine

2.4.2 Advantages and Disadvantages

The use of boundary scan test method has some potential benefits that allow us be confident regarding coverage when using BSCAN. Time needed for testing is now shorter while test coverage is higher. The use of boundary scan cells also contributes to the increasing diagnostic capability. The use of the BSCAN cells makes the test development simplified/automated.

However, the main drawback when using BSCAN is that there is an overhead in area since we use more hardware component in order to implement the technique in this case. More specifically, the registers and the boundary scan cells and the multiplexers are mainly the area overhead reason and as a result they probably dissipate more power energy. It should be mentioned though that technology evolution makes this drawback less significant over time.

2.5 Functional Testing

2.5.1 FT Description

Functional Testing (FT) is usually the last test strategy used before shipping the manufactured product. The result of the FT is PASS or FAIL and determines whether the product will be given to the market or not. PASS means that the product does not have any defect that could affect the functionality of the product. Otherwise FT gives us a FAIL indication.

In general FT is responsible for functionality verification of a PCB. In addition to this, FT procedure can be different between different PCBs. Functional testers are usually connected to the PCB under test through its edge connector or a test-probe point.

The most common type of functional test is called ‘’hot mock-up” and it mainly verifies that the PCB is functioning properly. Some more sophisticated functional tests are passing the PCB through an exhaustive range of operational tests. [12]

In general FT considers two approaches when it comes to faults: structural and functional faults. For example if we want to perform at speed test on a serial link we test for both functional faults (since we test at speed functionality) and structural faults (interconnects related faults).

2.5.2 Advantages and Disadvantages

Customer Advantages of Functional Test [12]:

- Functional test simulates environment for the product under test thereby minimizing the expensive cost for the customer to provide the actual testing equipment.
- With FT we do not need expensive system tests in some cases, which help the Original Equipment Manufacturer to save lot of time and financial resources.
- FT can give us the best possible result with careful handlings. That makes FT the most effective tool for system testing.
- Functional test gives us better results when it is combined with other types of testing such as ICT and FPT (flying probe testing). In this case it makes the product more robust and error free.

A functional test simulates the operational environment of a product in order to check its correct functionality. The operational environment includes any component that is connected to the Device under test (DUT) such as the power supply.

During functional testing the board is functioning and the functionality verification comes through the responses that we check at the checkpoints. The main advantage in this test strategy is the detection of faults like functional failures and other parametric failures. In addition to this, functional testing can also be run automatically by combining software with hardware components (eg. I/O boards and communication ports)

On the other hand the major drawbacks of the functional testing are the following:

- We have to perform fault estimation in order to be sure about the produced fault coverage.
- The functional test report contains information regarding measurements in a format that does not really help a lot the user to use it against defects.
- The Fault diagnosis usually gives to the user a PASS or FAIL status which is not so informative statement. Although FT can verify that the product is “fit for purpose”, it is not the ideal strategy for detecting manufacturing process related faults, often leading to a large number of boards within repair environments.

As soon as the functional test is over, the user has to perform various calculations in order to estimate the defect coverage.

2.6 Combination Strategy

From reference [19] we find useful information for combination strategy. Agilent Technologies has developed a software tool for Functional Testing achieving in this way the contradictory goals of increasing product quality and lowering costs. It simulates functional test and then generates metrics that are actually measuring the effectiveness of this tool. The PCOLA-SOQ FAIM standard is used.

FAIM is an extended version of FAM standard, regarding Functional Testing. Extension 'I' represent in parallel testing functional property.

The tool also enables the user to observe test gaps and redundancies as the following figures show.

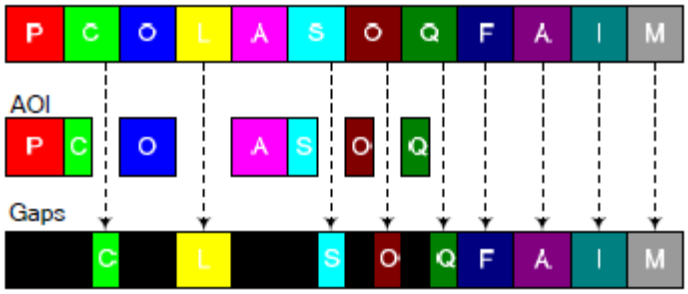


Figure 9.Test Gaps Identification

The test gaps are the light shining gaps pointed by the arrows. The above figure is an example of AOI test method which is of course irrelevant to the Functional Testing but it is used as an example to demonstrate the test gaps that can be formed when using a test method.

A more clearly stated example of the test gaps that can be formed when using different test methods is shown below:

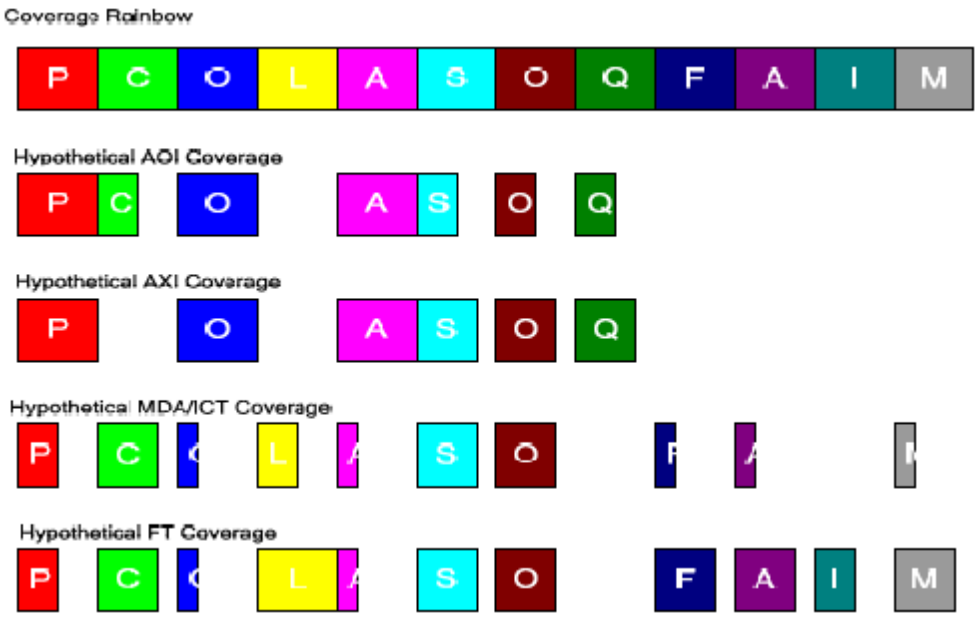


Figure 10.Combination Strategy and Test Gaps I

From the above figure we notice that every test method does not check fully all the properties of the PCOLA-SOQ-FAIM standard. In every case there are some missing parts for every property which are called the test gaps. In case of functional testing we observe that the FAIM properties are quite well supported in contradiction to the structural oriented properties (PCOLA-SOQ) where there are quite many test gaps.

However if we combine all the test methods mentioned above (AOI, ICT and FT) we can get the minimum test gaps.

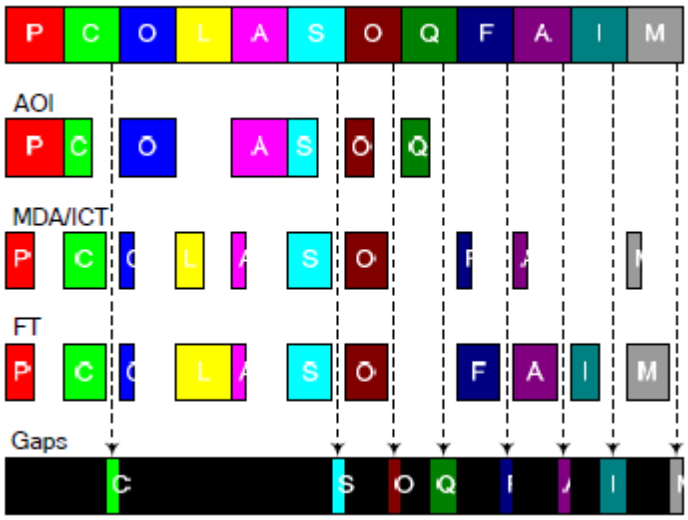


Figure 11.Combination Strategy and Test Gaps II

The test gaps are the lighted spaced at the Gaps bar. The Venn diagram of the defect universe could be the following one:

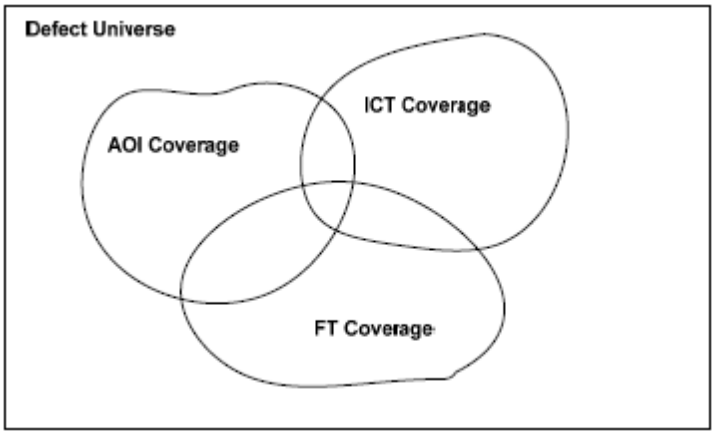


Figure 12.Venn Diagram of the Defect Universe

The test gaps that we observed before are basically the space of the defect universe that does belong to any of the three test methods of the Venn diagram.

3 Commercial Tools

Table 5. Commercial Tools

	ScanNavigator	ScanWorks	Provision	ScanExpress	D2B	Test Expert	TestWay
AOI	X	X	X	X	V	V	V
AXI	X	X	X	X	V	V	V
ICT	X	X	X	X	V	V	V
BSCAN	V	V	V	V	V	X	V
FT	X	X	V	X	X	X	V
FPT	X	X	X	X	V	V	V

3.1 ScanNavigator BoundaryScan SW Tool Powered by Victory

ScanNavigator Boundary Scan Test includes test, programming, and run-time modules that make simpler the test generation, increase efficiency, and allow full integration of test data, programming information, and diagnostic databases using Acculogic's testability and coverage analysis tools.

ScanNavigator Integrated Development Environment (IDE) is powered by the VICTORY boundary scan Automated Test Pattern Generation (ATPG) engine, and enables the user to select from a wide range of features such as data creation wizards, reusable test development and execution templates.

The ScanNavigator Run Time Environment (RTE) provides a wide range real-time control features to direct the flow of the BSCAN test program based on the pass/fail status at every test step. [15]

3.1.1 ScanNavigator's Access Analyzer Module automates pre-layout testability analysis

Access Analyzer is used after schematic capture and before CAD layout for assemblies that have a combination of scan and non-scan devices. Access Analyzer identifies all the scan nets and scan control nets (TDI/TDO interconnects) where physical test points can be omitted without risking the test coverage.[15]

3.1.2 ScanNavigator's Virtual Interconnect Test Module (VIT) offers 100% pin-level fault coverage

The ScanNavigator VIT module generates patterns, through circuit netlist and BSDL models, which are used to test BSCAN nets using only the virtual access provided by the BSCAN (JTAG) circuitry. On pure boundary-scan nets, VIT enables the user to get a full coverage of all open and shorts faults from silicon to lead bonds, from solder bonds to the circuit board itself. [15]

3.1.3 IEEE 1149.6 Support

ScanNavigator supports IEEE1149.6 (dot 6) Interconnect Test. IEEE 1149.6 and IEEE 1149.1 interconnect tests share the same design description, netlist and constraint definitions. The BSDL models are used in order to let the ScanNavigator detect automatically the dot6 device.

Scan Navigator considers the coupling capacitors as active components during AC testing and as an open during DC testing. In general ScanNavigator provides the following:

- Testing of AC-coupled and differential nets
- Automatic detection of IEEE1149.6-compliant devices
- Support of special IEEE 1149.6 features[15]

3.2 ASSET ScanWorks

ScanWorks BSCAN testing supports a wide range of 1149.x tools for testing boards and systems. Model based development gives the user the advantage of saving money, time and resources. The GUI provided for ScanWorks tests makes it easy for the user to understand and develop tests.

The speed of the tool is estimated to be in terms of minutes for any UUT or system. [16]

3.2.1 Design Browser – CAD and Schematic

The ScanWorks design browser is basically a part of the tool that is responsible for reading any Computer Aided Design (CAD) or schematic format from any vendor. It basically translates CAD/schematic file to a netlist and then calculates the fault coverage and automatically presents pin-points faults. [16]

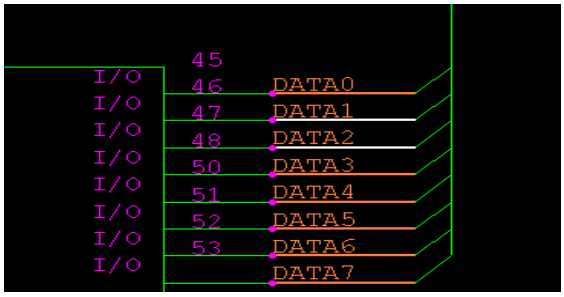


Figure 13. Schematic Figure

3.2.2 Interconnect Testing

Regarding interconnect tests, ScanWorks allows supports IEEE1149.1 and IEE1149.6 nets. [16]

3.2.3 In-System Programming/Configuration

ScanWorks Boundary-Scan Test can configure and program SPI (Serial Peripheral Interface) buses, FPGA's etc using various methods. The vector generation is configured in such a way that it automatically speeds up the process of system programming. [16]

3.2.4 Memory Test

ScanWorks can be used for testing memories such as DDRx, SRAMs, DRAMs or other kind of memories. It is a step by step procedure which basically allows the user to develop and test for instance DDR3 memories with a few mouse clicks. [16]

3.2.5 Powerful Debug Tools

Another important side of ScanWorks is that it has well customized and easy to follow debug tools in the sense that the user is able to understand and have a clear view of what the debug tool is doing at that moment. [16]

3.3 JTAG Provision

From reference [6] we find the datasheet of JTAG Provision. JTAG Provision enables the user to test a PCB reliably achieving in the same time the best possible fault coverage for faults that may show up during PCB assembly.

3.3.1 Automatic Test Generation

Automatic Test Generation is an important feature of JTAG ProVision that can be enabled by giving as an input the PCB's CAD derived netlist and BOM along with the ProVision's built in model library with the help of some powerful algorithms.

Automatic Test Generation includes interconnect tests (including 1149.6 test), memory cluster tests and logic cluster tests. Those tests combined together cover the majority of the connections on the board.

3.3.2 Developing Additional Tests

Developing Additional Tests is something that JTAG Provision considered as an add-on feature for cases where some connections cannot be covered by automatically generated tests. For instance connections from/to A-to-D and D-to-A converters cannot be covered.

Next it is up to the test engineer to develop additional tests using ProVision's Python JFT (JTAG Functional Test) scripting routines. Since JFT controls all the low level details of BSAN test, test engineers can focus on the development of the actual tests and in particular what data needs to be applied to which pins to find potential defects in the specific connections.

3.3.3 Fault Coverage

ProVision can generate the fault coverage of the automatically generated tests plus any additional tests. Moreover ProVision calculates in the first place the board's potential testability (how many faults can be detected). As a result the test engineer can later compare the testability against the actual fault coverage to check whether additional tests are required or not.

3.3.4 Use of Connectivity Data

ProVision allows the import of various netlist types mainly from schematics but also from board layout or computer aided manufacturing tools. Combining the connectivity data with component models and BSDL (Boundary Scan Description Language) test description models it is possible to get from JTAG ProVision an early estimation of fault coverage using BSCAN. The software that ProVision uses, gives us a predicted and an actual coverage.

The predicted coverage report is generated after processing netlist(s), device models and BSDL (Boundary Scan Description Language) models and is based on some assumptions regarding I/O access to connectors and test points. Actual Coverage is generated after performing an analysis of separate pcb tests that include scan path infrastructure, bscan-to-bscan pin interconnects, memory tests and logic cluster tests.

3.3.5 Schematic viewing and probing

ProVision offers a plug in module that basically highlights the fault coverage differences using a color coding system that helps the user to easily understand which pcb sectors are being tested fully and which need to be tested further.

The most important advantage that JTAG Provision offers from a HW engineer's point of view is that he can switch directly from the schematic page to the boundary control software thanks to the interactive schematic viewers.

3.4 ScanExpress DFT Analyzer

Scan Express DFT Analyzer is an automatic test coverage analysis tool for PCBs that contain both BSCAN and non-BSCAN devices. The tool also gives the ability to the user to increase fault coverage and reduce BSCAN test development time. Scan Express DFT Analyzer generates a number of test

coverage reports that includes all the pins, nets and devices classified into categories depending on whether they are fully tested, partially tested or untested (3 basic testability groups in total) and the user can view those reports by using the built-in report browser. [25]

The report also recommends where to add physical test access points in case we want for example to perform ICT (in-circuit test) in order to improve the fault coverage.

3.4.1 Features

- Generates test reports at the device, net, and pin level
- Generates both detailed test reports and test summaries
- Identifies incorrect compliance enable pin states
- Integrated 'Report Browser' interface
- Customizable analysis: include or exclude specific nets, buses, pins and devices
- Exports test coverage data to a spreadsheet or database for integration with other test equipment reports [25]

3.4.2 Benefits

- Establish exact boundary-scan test coverage
- Prepare test coverage reports for management
- Reduce the number of PCB test points
- Find test holes and deficiencies
- Increase fault coverage
- Reduce test procedure development time [25]

3.5 Teradyne D2B Strategist

D2B Strategist software is basically a tool that is used for test coverage analysis on PCBs and provides the test engineer with analytical reports. D2B Strategist software is trying to compromise the right amount of test, at the right cost, to produce the highest yield.[26]

3.5.1 Optimized, Cost-effective Test Strategies

D2B Strategist software is a tool that targets to optimizing test strategies, test times and in the same time calculates the effectiveness of a desired test strategy to find defects.

Strategist:

- Measures the actual coverage for a platform that is under test.
- Increases the testing performance by distributing the testing time between different test strategies.

Strategist GUI allows the test engineer to modify the analysis either by configuring various parameters or importing some of his own models such as Defects Per Million Opportunities data (DPMO). Then the tool measures the performance of the test by measuring test time and actual fault coverage for the selected test.

The resulting optimized test strategies are providing us with the highest fault coverage at the lowest costs. One of the advantages of using D2B Strategist is when it comes to complex products. In particular, in those products we may be provided with few physical test access points. However, Strategist gives the user the ability to maximize the fault coverage before the routing stage of PCB.

D2B Strategist software is a part of Teradyne's Design-to-Build suite of software tools that enables engineers to improve designs and provide the market fast enough.[26]

3.6 Unicam Test Expert

Simatic IT Unicam Test Expert is a tool that performs testing and inspection in a very good speed and uses a fast user-configurable nail/probe selection routine. It also generates automatically configuration files for more than 75 ICT, flying probe testers, x-ray and AOI machines. [17]

3.6.1 Library checker

Simatic IT Unicam Test Expert tool gives the user the ability to check and correct important parts of the BOM (Bill Of Materials) that may be missing. The part attributes are color coded whether they are optional or mandatory. [17]

3.6.2 Library editor

The Library Editor is used to add and modify properties for individual or selected electrical devices and parts. For instance, there may be problems that can be eliminated by using a graphics editor to correct polygonal body limits. Also, complex devices can be divided into simple equivalents in order to avoid the unnecessary creation of new complex test models. [17]

3.6.3 Probe selection/fixture design

User-configurable profiles include rules for automatic nail/probe selection on in circuit and flying probe testers. As a result, it is easier for the engineer to choose probes for single or panelized boards (boards that contain multiple instances of one circuit). [17]

3.6.4 Board accessibility report

The report states if no or not enough nail/probe locations were found on each net and indicates why. For example there may be nodes on a net that cannot be accessed due to soldering deficiency. [17]

3.6.5 Graphics links and search engines

There is a direct communication between the accessibility report and the schematics viewer that helps the engineer to view problematic nets directly by switching automatically from the report to the CAD graphics. [17]

3.6.6 Interactive nail/probe editor

The nail/probe editor is used in order to check the nail/probe placements and configurations in case they are changed. Such a case would be when there are very specific requirements and the test engineer has to change (move or delete) some nails/probes. In such cases, the nail/probe editor is verifying those changes according to some rules. [17]

3.6.7 Testability report

This report provides the user with information about overall net accessibility/inaccessibility, net coverage, part coverage, probe details, probe quantities and a nail type summary all in a well-structured HTML report. [17]

3.7 TestWay

From reference [2] we can find useful information regarding TestWay functionality. TestWay gives the ability to the designer to validate the design at the schematic stage in order to make sure that the manufacturer's requirements are met. Moreover, test engineers can use TestWay in order to estimate coverage value for various test strategies and in the same time the tool can give some recommendations to the user on how to improve the coverage.

The tool reads the PCB netlist (schematic or layout) and the component model libraries and then it performs an analysis while checking at the same time certain rules regarding topology and functionality.

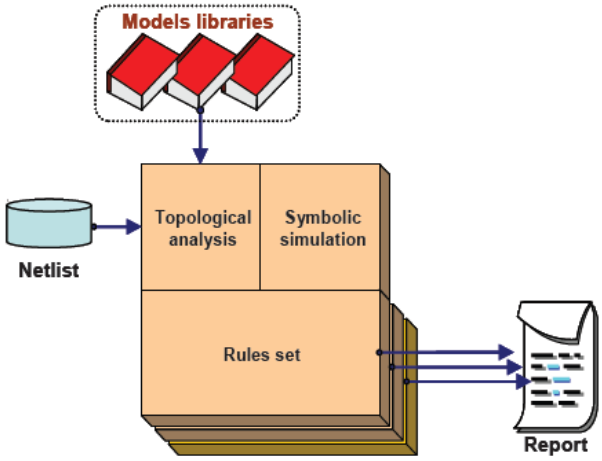


Figure 14. TestWay Functionality

3.7.1 Key product benefits:

- Design rules checking
  At this step the tool confirms that certain design rules have been used and implemented. In this way we achieve to avoid costly design errors at the earliest checkpoint.

- DfT rules checking
  This step is including verification of DfT requirements used for maximizing test coverage.

- Test point saving
  TestWay is capable of identifying nets that do not require physical test access. Instead, it places test points only where it is necessary thus reducing cost for test fixtures.

- Test coverage estimation
  Estimate coverage value for every test strategy.

- Test coverage measurement
  Determine real coverage against theoretical coverage.

- Functional test coverage
  Declare the parts and pins for FT and generate the relevant reports used for faulty board diagnosis.

- Board visualization
  Visualize test coverage in schematic and layout. Digitization procedure on schematics PDF creates an interactive schematic view.

- Advanced reporting
  Generate a variety of reports containing information such as fault coverage and production yield.

3.7.2 Rules Checking

TestWay's rules are divided into different categories:

• Design rules are used to verify restrictions imposed by certain technologies i.e. open-drain, specific termination requirements, noise immunity, etc.

• Boundary-scan rules are used for confirming boundary- scan compliance or identifying boundary-scan clusters.

• Custom rules to meet any specific company's requirements.

User can create new rules and import them into the analysis.

3.7.3 Test Point Saving

Nowadays PCB boards tend to have great complexity and as a result the number of nets in a PCB design is quite big. As a result, it is difficult to have physical test access to every net. That is why TestWay tool optimizes the number of locations where physical access is necessary.

3.7.4 Coverage Estimation

TestWay takes into account various test scenarios and simulates different test line combinations in order to estimate the overall coverage. It also sets the defect rates based on MPS (Material, Placement, and Solder) criteria, in format of DPMO (Defects Per Million Opportunities) for each component category. In this way it indicates to the test engineer the number of defects in the manufacturing process.

Test coverage estimation is done using a mix of available physical test access and any virtual test access provided by BSCAN cells. The important side of this procedure is that the coverage estimation is done during the schematic capture stage and as a result any improvements can be done before moving on to the next step (board layout).

There are many advantages when using TestWay as early as possible such as:
• Higher test coverage since any testability issues are defined prior to the design phase being completed.
•Improved test efficiency since the engineer can optimize test point placement and back annotate the test access requirements on to the schematics.
• Lower fixture costs due to optimizing the number of test points.
• Faster time to market since most of the DfT issues are confirmed and fixed during schematic capture level.

3.7.5 Coverage Measurement

TestWay can also measure the actual coverage by reading the actual "test programs" or coverage reports. Those "test programs" or coverage reports are generated by the corresponding test systems and are given as an input to TestWay. In this way we can compare the actual coverage value with the estimated coverage value, to ensure that we have met the original requirement.

Available coverage importers include HP3070, GR228x, Z1800, SPECTRUM, HP5DX, BST test systems, etc.

3.7.6 Functional Test Coverage

TestWay calculates functional test coverage using one of the following methodologies:
1. **Declaration:** Using schematic and/or layout files as an input mechanism for editing the functional coverage.
2. **Inheritance:** Test reuse in a hierarchical design flow where a functional block is associated with test coverage calculations. The functional test coverage report produced by TestWay is reusable in production to facilitate diagnosis of faulty boards.

4 Selection of the Tool

Table 4 is a quite explanatory table that highlights the selection process for selecting the appropriate tool for this project. Those are the most common and powerful tools that exist in the commerce. We notice that ScanNavigator, ASSET, Provision and ScanExpress provide only BSCAN (and FT in case of ProVision). That means that we cannot choose any of those test tools since we need to use a tool that can analyze a combination of the three test strategies that we are interested in.

Teradyne is a powerful commerce tool for testing PCBs since it can support all the test strategies apart from Functional Testing. Unfortunately due to that fact we cannot rely on Teradyne even though it offers 2 out of 3 test strategies. Simaticit also misses test strategies and in particular it misses BSCAN and FT.

Moreover D2B tool is a post-layout analysis tool that can only be used once the user has committed to layout. On the other hand, TestWay tool allows coverage estimation pre-layout (at the schematic capture stage).

We can see that TestWay tool, from Aster Technologies, seems to be the most powerful tool that enables the user to analyze any combination of test strategies using the same tool every time. That is a very important factor when deciding to use a specific tool for assessing all of the issues relating to insufficient testability, optimizing and maximizing test coverage and ensuring there are no gaps in the overall test strategy.

Moreover, using more than one commercial tool is much more expensive than using one and that is a very important factor when it comes to financial aspect of choosing a tool. As a result, from a financial point of view it is reasonable to claim that the TestWay tool is the most advantageous tool from a financial point of view.

From a technical point of view it is a fact that when a commercial tool can analyze a wider range of test strategies then it is sure that the support department is larger. As a result there are more expertized people in the same department which means faster response and better collaboration within the provider's company and that means better service.

Taking all these key points into account it is clear the fact that TestWay tool from Aster Technologies is the most promising and suitable testing commercial tool in comparison to all the others available.

5 TestWay Evaluation

5.1 Input Files

The board that will be used under BSCAN, AOI and FT test strategies is TCU. The files that were needed to be given as an input to the ASTER tool for those test strategies are the following:

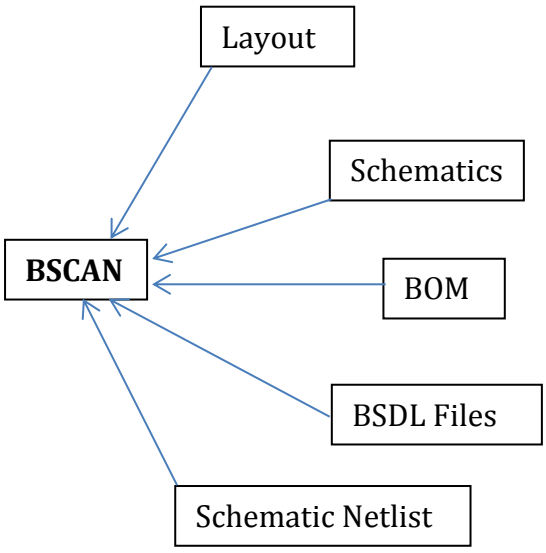


Figure 15.BSCAN Input Files

Layout: We need to import the Layout File of the PCB to the TestWay tool. The Layout file is in xml format and is provided by the PIWin database of Ericsson.

Schematics: We need to import the Schematics File of the PCB to the TestWay tool. The Schematics file is in PDF format and is generated by the Allegro Design Entry HDL tool.

BOM: The Bill Of Materials need to be imported to the TestWay tool. The BOM is in .csv format.

BSDL Files: The BSDL (Boundary Scan Description Language) files are necessary since they describe the boundary scan device package, pin description and boundary scan cell of the input and output pins. Without the BSDL files it is impossible to perform a BSCAN testing.

Schematic Netlist: This pstxnet.dat file contains information regarding all the nets, pins and pin vectors. It is important in the first phase of the project creation since it is helping the tool to identify inputs and outputs in all the parts.

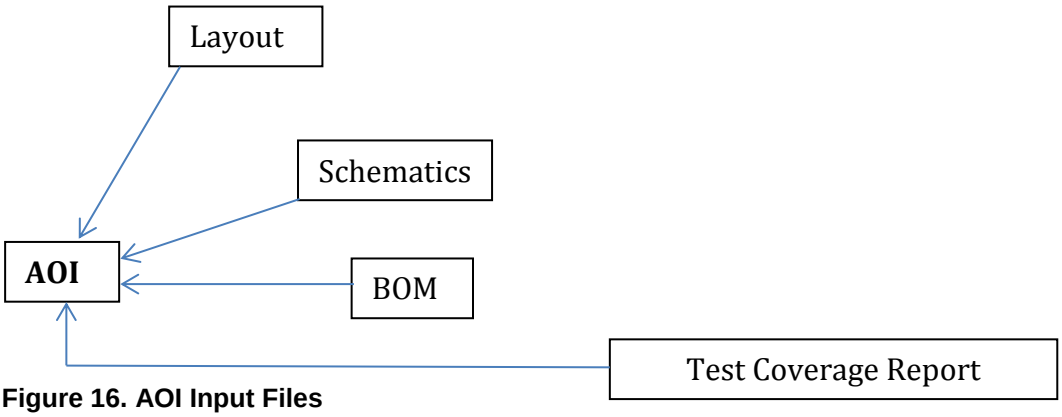


Figure 16. AOI Input Files

Test Coverage Report: The Test Coverage Report is a .csv input file that contains the AOI coverage information that was produced by the Orbotech S22 model and is used for the real AOI coverage. For the theoretical we simply need to model the capabilities of the target AOI machine.

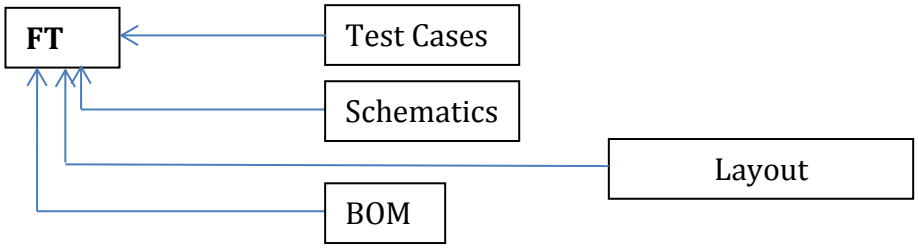


Figure 17. FT Input Files

Test Cases: The test cases are testing and checking whether specific parts of the PCB board function properly.

5.2 BSCAN Evaluation

5.2.1 Project Creation

After opening TestWay we create a New Project named PBA. [Figure 18]

Then we need to import a netlist input file and fill in the Project Name and choose Project Template. The netlist file that we will use is pstxnet.dat that we got from Allegro DE HDL tool. The Project Name will be PBA. Since it is about BSCAN we will choose bst_analysis.prj as project template. [Figure 19]

Next we will configure the tool before we run Analysis for first time. By clicking on Models/Unknown Models we choose "stop". [Figure 20]

We will also need to configure a parameter in the tool regarding ROOT NAME. By clicking on Analysis/Other Parameters we will set the ROOT NAME parameter. [Figure 21]

The ROOT NAME basically shall be given the same name with the Project's Name that we assigned in the previous step.

For this step, we will use the Schematics Netlist file (that contains all the models of the PCB). After analyzing we will see that the tool recognizes a number of unknown models (91 in total) and that is expected since the schematics netlist file may contain device types that may not be recognized by the tool.

For this reason, modelling is the phase where the user will give attributes to those unknown models. But before modelling we need to be able to locate those unknown parts to a graphical viewer. TestWay gives the user the ability to create a QuadView Project that basically is a graphical viewer that lets the user to visualize the schematics of the board.

5.2.2 QuadView

In order to create the QuadView Project we need to import to the tool the pdf schematics file and the TWL file. TestWay can produce its TWL file during the analysis in order to get the required information in its own format.

After importing those files we will be able to see the schematics in a graphical viewer. Another important characteristic of this point is that the tool can digitize all the parts in the schematics so that it can locate those parts every time we want to track one from the reports (by clicking on them).

5.2.3 TestWay BOM Creation

Now we will import the BOM file of TCU board to the tool. An important issue that needs to be noted at this step is the number of not-populated devices. The CAD data that we imported to the tool in the beginning (pstxnet.dat) contains devices, pin descriptions, etc that are included in the PCB design. However it can be the case that some devices from the CAD data are not included in the BOM data. Those devices are listed from TestWay as Not Populated.

It could also be the case that those devices are included in the BOM but still be listed as Not Populated. That is probably due to the fact that they are given a "NO" value to the "POPULATED" attribute in the BOM file.

We will create the TestWay BOM file in order to verify that we have imported the required information into the TestWay database.

5.2.4 Model Creation and Import

The Analysis includes a list with a number of unknown models. As we can see from the picture below there are 91 unknown models identified by the TestWay tool. Those models need to be implemented somehow. TestWay provides us with the ability of creating those models and import them to the project. [Figure 22]

Firstly we need to notice that there are two BSCAN devices in the list with the unknown models. Since we have the BSDL files (that include all the information regarding those BSCAN devices) we can import the ready models of the BSCAN devices in the first place and then let the tool create and import the models for the rest cases.

We already have the BSDL files for the following BSCAN devices:

RYT123153/1-RYT123153/1_CPM

And

RYT139652/1V2-RYT139652/1_5SGXB

So in the first place we notice that there are 91 unknown models and we can identify the first BSCAN device in the list (D1A2).

Now we will import the BSDL file for this BSCAN device and then we will re-run Analysis and we should expect 90 unknown models instead of 91. In order to import the BSDL file for D1A2 BSCAN device we first need to create a library that will be used only to store the BSDL files.

So we click on File/New/Library and then we press the  icon in order to import the BSDL model (.des file) and then we save this library at the Project directory as: PBA_BSDL. We can see now the details of the imported BSCAN device. [Figure 23]

One more thing that we need to do before running the Analysis is to map the imported BSCAN device name to the cross reference file. We notice that the name of the imported model is X7 while the name of the unknown model is RYT123153/1-RYT123153/1_CPM.

We need to map in some way the imported BSCAN device so that the tool can link the unknown model RYT123153/1-RYT123153/1_CPM with the imported BSDL file. In order to do that we open the cross reference file PBA.xrf and we simply write the following command:

RYT123153/1-RYT123153/1_CPM -> X7

Then we run Analysis once again (F5 key) and we can see now that the number of unknown devices is now 90 as shown below. Notice that the BSCAN device is no longer in this list since it was imported just before. [Figure 24]

After doing the same for the other BSCAN device we will create and import all the other unknown models, since they are not existent. In order to do that, we need to change the configuration of Models/Unknown Models from “stop” to “create and continue” since we want to create the models for the unknown models.

Then we run Analysis and we notice that there are no more unknown models and all the created models are imported to a new library called PBA.des. [Figure 25]

5.2.5 Modelling

Then we need to move on to the Modelling phase. During Modelling phase we check and correct classes and pin descriptions automatically assigned by the tool to the imported/created models in PBA.lib. [Figure 26]

When we try to compile the PBA.lib library we will see that there are 7 errors. There is a window that shows us what errors exist and by double clicking on them we are transferred to the correspondent faulty model.

For this model, we see that it is a gate while the model attribute has been automatically given a Tristate class. So we change the class like the images above demonstrate. We also notice that pin 3 (Ground Pin) has been automatically given a Power 1 Value which is wrong. That is why we correct by selecting Power 0 Value. That is the way we correct models during modelling phase.

A very useful attribute is the SPLIT attribute that basically allows the breakdown of components to sub elements such as resistor or diode arrays. For example a NFET can be modelled as n FET units combined into one unit.

Another important feature that TestWay provides us is that we can immediately switch from module specifications to the correspondent placement of this module in the schematics. In order to do that we just need to press F4 key when checking a model in the library and then we are automatically transferred to the same module in the schematics. For instance we can do that with a transceiver. [Figure 27]

When we press F4 key while being in library what is happening is that the tool is loading the input files that we imported during creation of QuadView Project in order to open the PBA.aa schematic. Then the schematics open at the place where the transceiver is located and we can visualize the highlighted pins that were generated during the digitization procedure.

That is very useful especially when the user is not sure about the class of a module and needs to check the module instance in the schematics. In this way the user can identify the class by looking at the connections, pin names or something else that can help him based on his experience to define the class of the module.

Everytime that we do a change at a module instance in the library, it would be better to compile the library in order to make sure that the change we did is valid and is not more an error. In the end of the modelling procedure we will get a confirmation dialogue box. [Figure 28]

As soon as we reach this step we will start with the Boundary Scan Testing Configuration of the tool. In the first place we will configure the tool to perform specific rules. [Figure 29]

5.2.6 Design Rules Checking

After running Analysis we verify that all of the selected rules have been correctly implemented. The most important that will be covered are the following:

- List of floating inputs
- List of floating outputs
- List of stuck inputs
- Check JTAG chain
- List BSCluster Candidates

Floating inputs and floating outputs are quite interesting information that the tool can provide us in order to notify the designer to correct the design properly. [Figure 30]

Those pins are not connected to any component and they have an NC (Not Connected) attribute in the library usually. However this should not be the case and instead those pins should be pulled up or pulled down through a small resistor. In order to change that, the designer needs to do modifications to the layout.

Next we check the stuck inputs. [Figure 31]

Once we click on the blue-colored pin name, eg. Pin D1101A2-1, from this list we will be transferred to its placement in the schematics. [Figure 32]

The tool notifies us about the exact location of that pin and shows us exactly the reason why this pin is in the list of stuck inputs. It is clearly shown that it is connected directly to the ground which should not happen.

The reason is that this pin is an Enable pin and should not always be “low”. Instead it should be high or low depending on the functionality (whether we want to enable or disable the correspondent component) that we need every time.

This list gives us in general all the pins with CMD (command) attribute that should not be directly connected to power or ground and should instead be connected to a pull up or a pull down resistor. However, again that is something that needs to be taken care of by the designer.

One of the most important issues that need to be checked is the “JTAG Chain” issue. It could happen that an input pin of a BSCAN device (in this case we

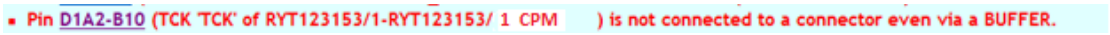
have two BSDL devices: RYT123153/1-RYT123153/1_CPM and RYT139652/1V2-RYT139652/1_5SGXB) is not connected to a connector. In this design there are two JTAG connectors:

X9801 (RPV380250/005-RPV380250/005-RPA)

and

X9900 (RNV445011/580-RNV445011/580-RNA)

TestWay let us know about the available JTAG connectors through a list of JTAG connectors that is generated every time we run the Analysis. However, it can be possible that some pins of a BSCAN device are not connected through a path to a connector. For such cases TestWay provides us a list with the pins that are not connected to a connector and details regarding the path that we could use in order to implement such a connection. For example,



That means that pin B10, which is basically TCK input of the BSCAN device, is not connected to any connector. In this case we need to find a path that connects B10 pin with one of the above mentioned available connectors. In order to do so we can click on the blue colored name of the pin ("D1A2-B10) and then we are automatically transferred to the schematics window at the page where this pin is located (PBA.aa file). Then we will press the  symbol in order to switch to the Netlist Navigator that gives us a better view of the pin under test.

As soon as we switch window, we will try to find a path that connects this pin to a connector. If we scroll down we will see that pin B10 is connected to R33A2 resistance.[Figure 33]

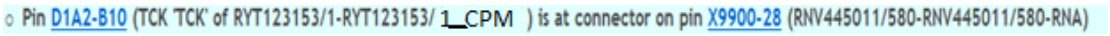
In order to find a path we click on the component holding the shift key in the same time so that the path is expanded from that component to the next component. We notice that at some point we reach at a buffer which means that the input is propagated to the output. So it is a component that needs no internal routing (in order to define a path through that component) as it could be for a multiplexer for instance.

We keep on expanding the path like the above images show until we meet the D9814 multiplexer. In this case we need to set an internal path through the multiplexer. So we need to decide which input of the multiplexer (pin 3 or pin 1) is going to be connected to output pin 4. If we examine the expanded path from pin 1 we will see that we will never end to any connector.

On the other hand, pin 3 is connected to X9900 connector. That means that we have to connect pin 3 with pin 4. In order to do that we open the test strategy file that is located under the Strategy directory in the Project Navigator (Strategy->bst.ts) and we write under section 2 the following:

Pin D9814-3 -> Pin D9814-4

That means that we choose to connect pin 3 of device D9814 to pin 4 of the same device. We shall notice that the direction of the assignment is from input to output. Now if we run Analysis we will see the following notification:



That means that the JTAG path is now recognized from the tool. In the same way we correct all the other JTAG paths.

The next list that TestWay provides the user is the list of BSClusters candidates. A Boundary–Scan (BSCAN) cluster is a single or a group of non-compliant Boundary Scan devices surrounded by Boundary Scan cells, which can be tested using a BST test strategy. The minimum access required is through a single test interface bus comprising of 4 test access signal pins (TDI, TDO, TMD and TCK), plus one optional asynchronous reset pin (TRST).

TestWay lists the devices that are candidates to be tested as boundary-scan clusters such as SDRAM, FLASH, and BUFFER devices etc. It also details the percentage accessibility of pins connected directly to boundary-scan pins.

Devices that have 100% accessibility to boundary scan pins are automatic candidates for boundary-scan cluster testing, whereas devices with less than 100% direct accessibility will require further analysis to consider whether access to these pins can be propagated.

It is very interesting to check the BSCAN cluster List that TestWay provides us. [Figure 34]

The yellow colored devices cannot be considered as clusters since they are not surrounded by JTAG cells. The devices that are surrounded by 100% of JTAG cells are automatically considered as BSCAN clusters. However, the difficult part is the devices that are surrounded by a smaller percentage of JTAG cells. In this case we need to examine the connections of the pins that are not JTAG pins in order to check whether they are connected or not to another JTAG pin through a path.

The most important feature that we are interested in the end is the BSCAN coverage value. In this case, after running the Analysis for first time, we notice that the BSCAN coverage value is 3.92%. [Figure 35]

However, this is not the best coverage value that we could achieve. There are still two more ways to improve this value.

We need to tune the boundary-scan test strategy to emulate the test capabilities that will be utilized by the target BST tester. Such as the use of parallel test connector access in order to enhance the test coverage, the consideration of interconnected non boundary-scan devices for cluster testing, and whether the boundary-scan devices are 1149.6 (JTAG standard) compliant in order to consider AC-coupled testing.

After making those changes we run Analysis for second time and we will notice in the end that the coverage value is increased. Not significantly, but it improved. [Figure 36]

The next step that we can follow in order to increase the coverage is based on the cluster candidates. We need to check which cluster candidates can actually be declared as BSCAN clusters. In order to do so, we open the Device Test Improvement Section of the Analysis home page and there we can see all the BSCAN cluster candidates that we saw before in the List of BSCluster Candidates.

In order to determine whether device pins can be considered as propagated access, or require physical access through a connector or a bed-of-nails probe, we will need to check each of the steps within the Device Test Improvement section and decide whether any of the devices in the list are cluster candidates. For instance TestWay says that access to pins 3 and 4 of device D9810 shall be considered as propagated access. [Figure 37]

Looking at pin 3 we try to find the path that leads to a JTAG pin. [Figure 38]

Then we extend the path to the next component. [Figure 38]

We notice that pin AN31 has the same direction with pin A4 for example. A4 is a BSCAN compliant Pin and that means that there is a path from AN31 to A4 and this means that access to pin 3 of Device 9810 can be considered as propagated access through pin A4. The same way we confirm the propagated access to pin 4.

We use the Interconnect Test Analysis Report in order to identify the nodes where boundary-scan access can be considered as propagated.

We run Analysis for third time and we notice that the coverage is further increased. [Figure 39]

5.2.7 Reports

The analysis of the BSCAN coverage is conducted according to the PPVS model.

The PPVS methodology makes it possible to take into account the precise coverage by providing different complementary test strategies, so that any duplication in coverage can be removed in order to provide an optimized test strategy with the highest possible coverage.

TestWay provides two template reports in the project navigator space under the Crystal Reports directory. [Figure 40]

The normal PCB assembly process is complex and has many places where something can go wrong. Every defect can be modeled into a fault-effect.

Depending on the test method selected, tests can be generated in order to check every fault-effect. This is called fault-oriented testing.

The test quality measure called "Fault coverage" defines the ability of one test method to detect some types of fault. In order to compare and combine "fault coverage" from various test methods, TestWay creates 4 coverage metrics: w Presence of a component w Polarity (orientation) of a component w Value (proper value of the discrete component, signals propagating through a part of the circuit) w Solder (Connections of the component to its surrounding).

These 4 metrics could be combined from structural test (AXI X-Ray, AOI Automated Optical Inspection, MDA Manufacturing Defects Analysis, ICT In-Circuit Test, BST Boundary-Scan Test) or functional test.

Figure 41 presents in a 3-dimentional view, the number of components and the different coverages per component complexity. Coverages are described in 5 categories: accessibility, presence, polarity, value and solder.

Figure 42 presents the coverages (presence, polarity, value) by partnumber per component complexity. For example, if at least one partnumber is covered in value, we can deduce that the right component is in the feeder. Then all parts using this partnumber are at least partially covered in value.

Figure 43 presents the covered solder joints versus the total amount of solder joints per component complexity.

Figure 44 shows the PPVS coverage for each device and the weighted coverage value, which is balanced against the defect opportunities as defined by the material, placement and soldering defect rates expressed in PPM.

In this example there are two connectors of type RPV 380 250/002. We notice that the weighted coverage value for the first one is 0% and for the second one is 84.62%. So in total, the weighted coverage value is 42.31% and the PPM for this type is the sum of the PPM value for each instance of this type.

We can also see the percentage of the covered solders/ total number of solders for this type of component. In this way we can see for every component this characteristic feature.

Now if we open the production model file under the crystal reports directory in the project navigator we will see the Total Production and Test Model. [Figure 45]

This is a very interesting figure because we can have a more general and clear view of the total Production. In particular we can see that TestWay provides us with a chart indicating the predicted yield figures based on the selected test strategies and industry standard DPMO figures.

Although this provides a relatively high First Pass Yield figure of 94.17%, the main area of concern is that 22.06% of faulty boards that are shipped to the customer. The figure needs to be reduced.

5.2.8 BSCAN TPQR Coverage Report

If we click on the 20.05% link above the BST image [Figure 46] we will be transferred to the TPQR Report. [Figure 47]

If we navigate to the individual integrated circuit coverage report by selecting the **Integrated Circuit** heading within the **Device Type** column we will be transferred to the Testability Report for Integrated Circuit. [Figure 48]

If we scroll down we can see some more tests. [Figure 49]

The "COMMENT" column of the "Integrated Circuit" table defines two types of boundary scan test:

- **JTAG test** These devices are tested as pure interconnects between compliant devices
- **Cluster** These are not-compliant BSCAN devices that are tested as boundary scan clusters.

Another interesting feature of TestWay tool is the ability to visualize coverage within QuadView. In particular we can click on a device from the list in the above image and then we will be transferred to the Schematic view where we will need to enable the "COVERAGE(BST)" option and then we will notice that there will be some green and red spots. The green spots declare that the correspondent pin is tested for solder otherwise there shall be a red spot. [Figure 50]

5.2.9 Estimated Vs Real BSCAN Coverage

If we want to compare the estimated BSCAN coverage value that we got from TestWay tool with the real BSCAN coverage value (which is achieved by the ASSET ScanWorks HW platform) we will need to give as an input to the TestWay tool the XML Test Coverage Report (CombinedReport.xml). Any ASSET ScanWorks 3.6 or later version platform will be able to produce this xml file.

As soon as we have the .xml file we can import the asset test strategy file (asset.ts) that is provided by TestWay tool and all it does is to load the .xml file and process the information of this file in order to provide the user the ASSET BSCAN coverage value and relevant information in a user friendly format. We can import the .ts file under the Strategies directory in the Project Navigator. [Figure 51]

Moreover we can compare the estimated against the real coverage by using a specific script file that is provided by TestWay tool. The script is called "compare_test_strategies.scr". [Figure 52]

After importing those two files we will run again the Analysis and then we will see the coverage values under the TPQR Report. [Figure 53]

The real coverage is really low and that is mainly due to the fact that there are only two JTAG devices and the specific board is a very complex board.

Moreover there are pins on clusters that are not accessed by BSCAN device and that means that they remain uncovered.

In addition to this, the net access is very low (1.86%) thus making it difficult to check all interconnects. If we click on the 1.42% figure that we see above we will be transferred to the test report for the ASSET tool and there is the 1.86% net access value.

TestWay also provides us with a comparative report that compares the coverage values (based on PPVS model) between ASSET tool (real coverage) and TestWay BSCAN coverage (theoretical coverage). [Figure 54]

Figure 55 is showing a part of that report.

We can see that the report contains coverage_presence_ASSET and coverage_presence_BST. Those are the BSCAN coverage values that are derived from the ASSET tool and TestWay tool in respect.

There are more attributes that are included in this report regarding coverage (such as coverage_polarity, coverage_value, coverage_solder) for both tools (ASSET and TestWay). However due to space limitation there was no space to display all the other attributes.

Moreover we can see that the RAM devices are registered as clusters in TestWay but they are not recognized as clusters by ASSET tool. That is why the theoretical coverage regarding presence is zero and the real one is 100%.

5.3 AOI

We will perform an AOI strategy through AOI (theoretical coverage) and we will compare it to the real AOI coverage value that we got from the Orbotech S36 AOI machine. We will keep on working on the project that we have already developed so far. We will remove the bst.ts and asset.ts test strategy files and we will later add new aoi relevant test strategy files.

In order to perform an AOI test strategy (to measure the theoretical coverage value) we only need to add the aoi.ts test strategy file that is provided by TestWay. In order to import the real AOI coverage that was produced by Orbotech S36 machine, we will need the top and bottom side report files (.csv) that are generated during the Orbotech S36 function.

We will store those files in the working directory and then we will import in TestWay the s22.ts test strategy file that is processing those .csv report files in order to generate the real S36 AOI coverage value and all the relevant reports. At Figure 56 we can see the two mentioned test strategy files.

Before running Analysis we need to configure the aoi.ts test strategy in order to achieve the best possible AOI coverage value. At Figure 57 we see the parameters that are already changed.

The “Inspection” PreReflow setting will consider that devices are checked for Presence and Polarity, but not for Solder, whereas the PostReflow setting will also consider Solder. The “Side” and “Mounting_Techno” settings are self-explanatory in that you can specify if Top, Bottom and Both sides are checked, and THT, SMD and Both respectively.

THT: Through Hole Technology regarding electronic components involve the use of leads on the components that are inserted into holes drilled in PCB and soldered to pads on the opposite side either manually (placement by hand) or automatically (mount machines).

SMD: An electronic device that is made by a PCB and all the components mounted on the PCB using SMT (Surface Mount Technology).

However, it is unlikely that the optical character recognition feature (OCR), will either be available or working effectively. We will change the OCR setting to “No” and leave the other settings the same.

Now we will run Analysis and as soon as it finishes we will go to TPQR Report and there we will be able to see the theoretical and real coverage values. At Figure 58 we can see those values.

We can see that the theoretical AOI coverage value that was calculated by TestWay (48.18%) tool is very close to the real one (47.14%).

It would be very interesting to check some reports generated from AOI test strategy regarding theoretical and practical coverage. [Figure 59] demonstrates the theoretical coverage.

The report results are the same for real AOI coverage value. There are very slight differences which are not considered in the end.

5.4 Functional Testing

Regarding Functional Testing on TCU Board, we will select a few test cases (TC) from the test requirements document of TCU board in order to demonstrate how the FT is executed.

The main idea is to identify for every TC the components and pins that are taking part into this TC and capture them in order to get PPVS coverage results and FT coverage value in the end. This technique is called functional testing by declaration. The schematics figures that are attached in the appendix are used only in order to illustrate the green highlighted pins and parts that are covered for every test case.

5.4.1 Monitoring of secondary power supplies.

The purpose of this TC is to confirm that the power supply outputs are within some specific limits.

So we will have to perform FT on the Primary Converter (that is responsible for producing the 9V bus) and the secondary voltage producers.

For this TC we will demonstrate how we will capture the Primary Converter that is responsible for producing the 9V bus.

At Figure 60 we can see that we are on page 91 where the Primary Converter Circuit is drawn. We click on the test coverage button as shown and then we will create the test step for the Primary Converter as shown below just by clicking on the correspondent raw and typing the name of the test case. [Figure 61]

Then we click on the record button that the top arrow points to in order to capture the circuit of the primary converter that we want to test. After we finish the selection of the area (the circuit) we will click again on the record button (that is available on the toolbar) and then we go back to the Steps Setup Window (by clicking the test coverage button) and we save the test step. This is how we will also work for the BPM and secondary voltage Generators (Bucks).

When it comes to which components we need to cover for Primary Converter and Secondary Voltage Producers.

For the Primary Converter we will not need to cover the decoupling capacitors since they are used mainly for environmental disturbances (noise). We will also not cover the .TOM components since they are used during the development phase of the design and not for the functionality of the board. We will also ignore the unconnected pins (NC attribute). However we will cover all the IO pins of the Primary Converter. [Figure 62]

After covering the Primary Converter we will need to cover the components that are relevant to the Secondary Voltage Producers.

For example for the 5,5V generator we will need to cover all the IO pins of the regulator that is inside the circuit of the 5,5V generator. We will also need to cover the inductors and resistors that are connected to the pins and the voltage divider that is inside the circuit of the generator. [Figure 63]

The same way we cover the rest generators.

5.4.2 STATUS signals from the TUM

The test shall verify the function of the STATUS signals from the TUM. Those signals are:

Table 6. STATUS Signals

STATUS_AFREE_ICM[1]	Freerun PLL
STATUS_AFREE_ICM[0]	MULT PLL
STATUS_BRBS_ICM[1] (TCU)	TCU: 644.53125MHz BRBS
STATUS_BRBS_ICM[0] (TCU)	TCU: Synth BRBS
STATUS_BTN_ICM[1]	644.53125MHz B_TN
STATUS_BTN_ICM[0]	Synth B/C TN

First we locate the status signals in the schematics vewer and then we will try to capture the components (and their IO pins) that are relevant to those status signals.

The next table includes the pages and pins where we can find those signals.

Table 7. STATUS Signals-Pages at Schematics

Signals	Page/Pin
STATUS_AFREE_ICM[1]	26,28/GPIO<0>
STATUS_AFREE_ICM[0]	26,28/GPIO<1>
STATUS_BRBS_ICM[1] (TCU)	36/LD(25)
STATUS_BRBS_ICM[0] (TCU)	33/J17
STATUS_BTN_ICM[1]	40/LD(25)
STATUS_BTN_ICM[0]	37/J17

It is very important to notice that we want to cover only the relevant pins and components to this TC. For instance, regarding STATUS_BTN_ICM [1], that is produced from F_PLL component, we only need to cover the inputs of this component that only contribute to STATUS_BTN_ICM [1] output and ignore the rest pins. We will also ignore the components inside F_PLL that are irrelevant to the STATUS_BTN_ICM [1] output signal. [Figure 64]

If we go deeper inside the F_PLL component we will be able to see exactly which components and pins are relevant to the STATUS_BTN_ICM [1] signal and then we will cover them for FT.

We shall notice firstly that the STATUS_BTN_ICM [1] signal is connected to the STATUS_FPLL_ICM signal inside the F_PLL component. So we need to cover the STATUS_FPLL_ICM signal inside the F_PLL. [Figure 65]

So as we can see, inside the F_PLL component, the STATUS_FPLL_ICM signal is connected to the LD (25) pin of the Synthesizer component. So finally we can see that we need to cover the input pins of the Synthesizer that are

contributing to the LD (25) pin and the Synthesizer component itself. We can find which input pins are necessary to cover for LD25 pin by checking the Synthesizer's Manual.

At Figure 66 we can see the difference between the pin under test and the regular-uncovered pins of the synthesizer component.

The covered pins are highlighted with green color while the uncovered-unrelevant pins are blue-colored. We will work in the same way in order to cover the relevant pins and components for the other STATUS signals. We will also need to cover the devices that are connected to those pins (resistors and capacitors).

5.4.3 CPM to TAIPAN 1Gbit Ethernet Test

This Test Case is about testing the 6 interfaces in CPM component that are connected to 6 interfaces of TAIPAN component. Every interface has a transmitter (T) and a receiver (R) sub-interface. Every signal that crosses the path from a T/R to a R/T interface is differential. That is why there is a _P (positive) and a _N(negative) ending in the name of the relevant signals. The following figure is an example of one interface. [Figure 67]

So we will need to locate the differential signals that correspond to the T/R interface of CPM and TAIPAN.

The six T signals that are coming from the CPM component are the <3…1> signals of the HSS2_CPM_TX_N<3…0> and HSS2_CPM_TX_P<3…0> bus and the <3…1> signals of the HSS3_CPM_TX_N<3…0> and HSS3_CPM_TX_P<3…0> bus.

The six R signals that are coming to the CPM component are the <3…1> signals of the HSS2_CPM_RX_N<3…0> and HSS2_CPM_RX_P<3…0> bus and the <3…1> signals of the HSS3_CPM_RX_N<3…0> and HSS3_CPM_RX_P<3…0> bus.

So we will need to cover the <3….1> signals of those busses that are located on page 2,5,46 and 52.

HSS2_CPM_RX_N<3…0> and HSS2_CPM_RX_P<3…0> busses and HSS3_CPM_RX_N<3…0> and HSS3_CPM_RX_N<3…0> busses are basically I/O pins at the CPM1 block (on page 2) and we need to go inside that component in order to cover any device that may be connected to those pins.

For example, for pin HSS2_CPM_RX_P<3…0> and HSS2_CPM_RX_N<3…0> we will need to cover <3…1> signals. We go to page 5 where we can see inside CPM1 block and we will see that this bus is connected to a network of capacitors as we can see at Figure 68.

So, since we want to cover only 3-1 signals of the two busses we will only cover the capacitors that are covered with the red line and the 0-2 bits of the busses HSS2_RXSDP<0:3> and HSS2_RXSDN<0:3>.

So we do not want to cover pin 3 HSS2_RXSDP3 and its pin number is AY12.

In order to cover only 3 out of 4 signals of the bus we need to do the following:
-Record all 4 signals of this bus by recording the pin HSS2_RXSDP<0:3>

-Go to the generated list of the covered signals and delete the line that corresponds to the uninteresting pin (D1A2-AY12 for the above mentioned example). [Figure 69]
-Then run Analysis

We work in the same way for all the other busses that we mentioned above.

When it comes to the TAIPAN component we also need to cover the Transmitter and Receiver Interface. For the receiver interface we need to cover the SGMII_ITN_CPM_ICM<5…0> differential pin and for the transmitter we need to take into account the SGMII_ITN_ICM_CPM<5…0> differential pin.

So we need to go inside TAIPAN on page 52 and cover those pins as shown in Figure 70.

5.4.4 FPGA to TUM I2C Test

This test case is about verifying the two I2C interfaces between the TUM and the FPGA part of TCU board. In order to do so we need to locate the signals that correspond to those interfaces and cover also the blocks/devices that are relevant to them and necessary for the functionality of those signals.

These signals are:

- I2C_SCL_ICM_OCXO
- I2C_SDA_ICM_OCXO
- I2C_SCL_ICM_AFREE
- I2C_SDA_ICM_AFREE

The pages where we can find these signals are: 2,26,27,28.

On page 2 and 26 we can notice those signals connected to the OCXO_ADC and M_FR_PLL blocks on a higher level and that is why we need to go inside those components.

On page 27 and 28 we can manually record the parts that are very relevant to those signals. At Figure 71 we can see the inner architecture of OCXO_ADC block and the part that we covered (with green color).

We can see that we do not cover the capacitors that are connected to the reference supply voltage for SCL and SDA inputs since they are used for noise reduction and they do not really involve much to the functionality. They are tested, however, by using AOI.

Regarding the I2C_SDA_ICM_AFREE and I2C_SCL_ICM_AFREE signals we go to page 28 and cover the green highlighted compoents and pins that we see in Figure 72.

We can see that we also have a translator here like before (for OCXO component) and then we also cover the resistors that are included in the path till SCLK and SDATA pins of the Multiplier&Free Running PLL component.

5.4.5 SFP+I2C+ and control signals

This test shall verify the communication between the TAIPAN component and SFP connectors. In order to perform functional testing on this test case using TestWay we need to locate and cover the pins of TAIPAN that are connected to the SFP

connectors and the relevant devices that are taking part into the functionality of those I/O pins.

There are 8 SFP connectors in total and the signals that are propagated from TAIPAN to the SFP connectors are coming from the following busses: SFP_I2C_SCL<7…0> and SFP_I2C_SDA<7…0>.

In the first place we need to cover the : SFP_I2C_SCL<7…0> and SFP_I2C_SDA<7…0> pins. So we will need to go inside the TAIPAN block and capture those pins as shown at Figure 73.

So we will track the signals of those two buses. Then we will go inside the architecture of the SFP connectors and locate those signals and cover them with the tool. We will also cover the components that are affecting the functionality of those signals.

The first group of the 4 SFP connectors corresponds to the following signals:

SFP_I2C_SCL<7…4>
SFP_I2C_SDA<7…4>

The second group of SFP connectors corresponds to the following signals:

SFP_I2C_SCL<3…0>
SFP_I2C_SDA<3…0>

On page 55 we can see the first group of 4 SFP connectors on a high level. As a result we need to go inside those 4 SFP connectors. Pages 57 to 60 contain the internal architecture of the SFP connectors. We can see one of them, and the same configuration applies to all the other pages. [Figure 74]

5.4.6 DDR3 SDRAM Tests

This test is mainly about testing the interconnects between CPM and DDR3 memory devices and that correct memory devices are being installed.

There are in total 4 DDR3 memories : SMEM0,SMEM1 and CMEM0 and CMEM1. CMEM0 has the same architecture with CMEM1 and the same applies for SMEM0 and SMEM1.

So we will check CMEM0 and SMEM0 and the same configuration applies for the CMEM1 and SMEM1 components.

For CMEM0 we need to cover CMEM0 main memory, the CMEM ECC (erroc correction code) component and the CMEM Regulator. For SMEM0 we need to cover SMEM0 main memory and the SMEM ECC component.

For CMEM0 main memory we will not take into account the NC (not connected) pin and the Test Points. We will also ignore the decoupling capacitors since they will be covered by the AOI test method and they are not contributing to the functional side of the component.

When it comes to the SMEM memories we need to cover SDRAM0 and SDRAM1. SDRAM0 is exactly the same with SDRAM1 and that is the perfect case in order to perform the functional testing by inheritance.

For this technique what we actually do is that we cover the SDRAM0 components and pins that we are interested in and then we declare their equivalent components and pins in the SDRAM1 device. In this way we do not have to declare again the same pins and components for SDRAM1 device like we did with SDRAM0 device.

In order to do that we need to follow the next procedure:

1) Generate the TestWay Netlist for the SDRAM0 component.
2) Import that list in to the tool
3) Declare the SDRAM0 device as the original device that will actually be used in order to match its pins and components with SDRAM1's pins and components.
4) Run the analysis in order to make SDRAM1 device inherit the coverage data from SDRAM0.

In order to generate the TestWay Netlist of the SDRAM0 component we need to record at the schematics the SDRAM0 device (on page 5) and its internal circuit (on page 15).

Then we choose "Selection Mode" and we select those components and pins (that we just recorded) and then we go to Tools->Export Netlist in order to produce the netlist (TWL) file.

Then we import the TWL file by going to the Job Setup->Job->Netlist File. Then in order to declare the SDRAM0 as the original device we need to create a special type of file that has a .rul extension and by writing a small piece of code we can do that.

Then we just run Analysis and we can see that the tool considers the SDRAM1 pins and components as equivalent to SDRAM0 pins and components. [Figure 75]

The tool also provides us with report regarding FT by inheritance. For example we can see at Figure 76 that the report shows which pins and parts are considered to be equivalent.

5.4.7 Functional Coverage

In order to estimate the overall Functional Coverage we need to combine the Functional Coverage by Declaration with the Functional Coverage by Inheritance. In order to do so we just need to declare the combined functional test strategies and then run Analysis. As we can see in the figure below combining the two coverage values (coverage by declaration and coverage by inheritance) we get the combined-final FT coverage.

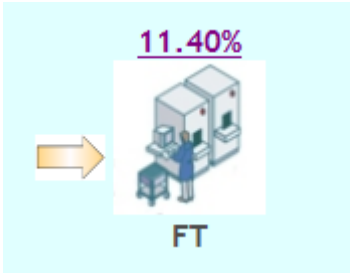


Figure 77. FT By Declaration

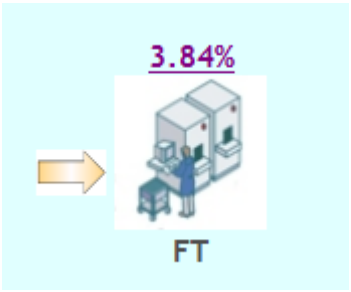


Figure 78. FT By Inheritance

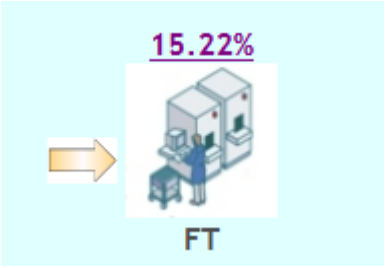


Figure 79. Total FT Coverage

5.5 Combined Coverage

5.5.1 BSCAN-AOI Combined Coverage

It is very simple to measure using the tool the theoretical combined coverage of BSCAN and AOI as soon as we have finished with BSCAN and AOI seperatelly.

Then we will need to import to the TestWay project the two test strategies files (BSCAN and AOI test strategy files) with the configurations that we have allready covered in the previous relevant sections and then we run analysis. In the TPQR Report section of the home page of the project we can see now the two coverage values that are combined. [Figure 80]

In order to check the combined coverage value we click on the purple link that is referring to the index.htm. But before doing that we shall notice that the AOI coverage value is higher when it is combined with BST strategy than the AOI coverage value that we got when we run the analysis just for the AOI itself.

The reason why this is happening is that the combination of those two strategies makes the AOI testing not to consider some devices that re tested by the other test

strategy. As a result the total number of considered pins and devices is decreased and therefore the total fault coverage is increased.

The following images show when the AOI testing is testing device D5000. In case of single AOI analysis we can see that device D5000 is covered but in case of combined AOI and BST the D5000 device is not covered by AOI (it is covered by BST though). [Figure 81, 82]

Next we can see that the combined coverage value is 75.27%. [Figure 83]

5.5.2 BSCAN-AOI-FT Combined Coverage

Now, if we combine all the test strategies (FT-BSCAN-AOI) and we run analysis we will can see alla the coverage values seperately at the TPQR Report section. [Figure 84]

We click again on the index.htm link and then we can see that the overall coverage is 77%. [Figure 85]

6 Conclusions

In this project the main goal was to select an appropriate and effective fault coverage analysis tool. After choosing TestWay tool the next step was to learn how to use the tool for BSCAN, AOI and FT test strategies and get in the end the fault coverage values. We also had the chance to compare theoretical and practical fault coverage value for BSCAN and AOI test strategies. In the end we combined the coverage from all the three test strategies and got the combined one.

It shall be mentioned that the coverage value that we got for FT is not a representative coverage value for TCU board. In particular this value is not the final FT Coverage value because it is the outcome of a small part of the test requirements document of TCU board. We chose a small number of test cases in order to demonstrate the functionality of the tool when it comes to FT. However, testing fully the TCU board (that means that we consider all the test cases from the test requirements document) will give us a much bigger coverage value and that would be the final one.

It should also be mentioned that Aster Technologies updated its TestWay Software tool in order to overcome some technical difficulties that occurred during the FT training period. In particular, it was necessary for the tool to recognize the busses in the schematics in order for the user to be able to cover those busses whenever a test case needed.

However the pdf schematics file, that the Cadence Allegro Design Entry HDL tool was generating, could not provide the tool with the necessary information regarding busses (such as bus attributes, bus sub-signals). As a result Aster Technologies updated their Software tool and then we could retrieve information about busses during FT.

It is also important to keep in mind the reports of TestWay tool and the fact that the user can easily locate which components are tested or not and directly locate those components at the schematics viewer.

However it is crucial to mention the effort needed when it comes to the FT especially, since FT is considered to be the most complex procedure perhaps bearing in mind that it is mostly a manual procedure. More specifically, the user needs to locate the relevant components and pins for each test case and cover them at the schematics viewer, as described in the relevant section of this report.

So the effort needed for FT using TestWay tool is depending on two factors. The first one is the user's HW design experience and the second one is the Tool's usability. For the first factor, the more experienced the user is the faster he can know the components and pins that he needs to cover for each test case. Regarding the second factor, TestWay offers a very easy and time efficient way of declaring components and pins as already described in the relevant section.

For this project, FT was performed on 6 test cases and the average time needed for understanding which parts and pins to cover for every test case was not the same for all of these test cases due to the variety of the difficulty. The following table shows the time needed for every test case.

Table 8. Estimated FT Analysis Time - Test Cases

Test Cases	Analysis Time(min)	Parameter	Estimated Time(min)
Monitoring of Secondary Power Supplies	20	12	240
STATUS Signals from the TUM	15	7	105
CPM TO TAIPAN 1Gbit Ethernet Test	20	6	120
FPGA TO TUM I2C TEST	15	16	240
SFP+I2C+control signals	10	27	270
DDR3 SDRAM Tests	30	1	30
Total Time	110	-	1005

The parameter column is an indicator parameter basically that shows us how many test cases of the same difficulty exist in the test requirements document of TCU board. In this way we can get an estimated view of the total time needed to perform FT for all test cases.

For DDR3 SDRAM Test the CMEM1 has the same architecture with CMEM0 and the same applies for SDRAM1 and SDRAM0. So that is why the time needed was not so much as expected, considering the number of the relevant to his test case components and pins.

This part was done in cooperation with Kim Petersén who has a lot of experience in this area and as a result the time used was the most efficient. The average time needed to declare the components and pins using the tool for every test case was 7 minutes (42 minutes in total for the 6 test cases). So for the FT part the average working time was 152 minutes. For the whole test requirements document it could be estimated about 1488 minutes (TestWay declaration effort: 69 test cases x 7 minutes/test case = 483 minutes). It is obvious that the time needed highly depends on the user's experience on the schematics and test cases of the board and less to the tool (since the procedure of declaring is easy and the same every time).

When it comes to time estimation for AOI and BSCAN it should be noticed that the time needed is different for BSCAN between boards with great difference in complexity level. Regarding AOI time is more or less the same since the estimation of AOI coverage is a very automatic procedure and the user only needs to insert the aoi.ts file and let the tool do the analysis.

Regarding BSCAN, supposing that the user has all the necessary files, the time needed is described in the next table.

Table 9. BSCAN Estimated Analysis Time

BSCAN Steps	Estimated Analysis Time(min)	Depends on
Project Creation-Unknown Models Creation-TWL Generation	6	2
QuadView-Digitization	15	2,3
TestWay BOM Generation	3	2,3
BSDL Devices Mapping	3	1,3
Modelling	90	1,2,3
JTAG Chain	15	1,3
BSCluster Candidates	20	1,3
Total	152	-

1. User's HW Design Experience
2. Computer's Speed
3. Board's Complexity

Table 10. Total Estimated Analysis Time for complex board

Test Strategies	Time(min)
BSCAN	152
AOI	7
FT	1488
Total	1647

So the total estimated Analysis Time needed for a complex board (assuming that we cover all test cases regarding FT) like TCU PCB would be 27,45 hours.

In general TestWay tool is a flexible tool that provides the user with all the important test strategies (giving also very explanatory and detailed reports) and in combination with the very well organized and cooperative support from Aster Technologies makes it an ideal tool for testing any PCB.

Further work for this project could be how to add more attributes when it comes to the PPVS defect standard. In particular a further study could be how to use the tool in order to import more attributes relevant to the FT that could give us more information about the FT part.(such as at speed test related attribute or any other attribute that could be related to functional faults).

Appendix

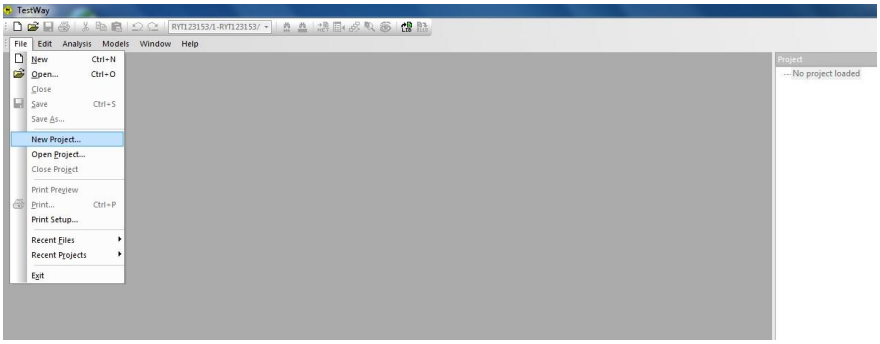


Figure 18. New Project

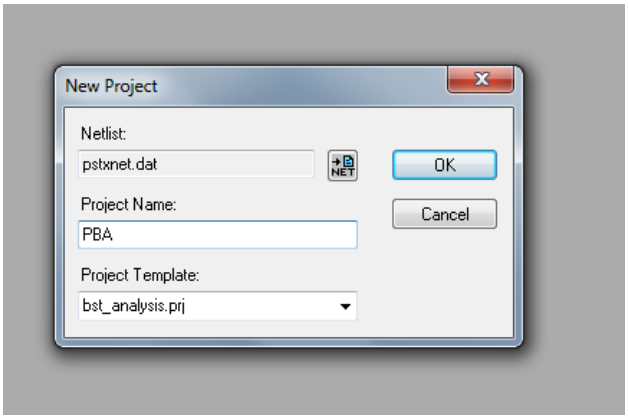
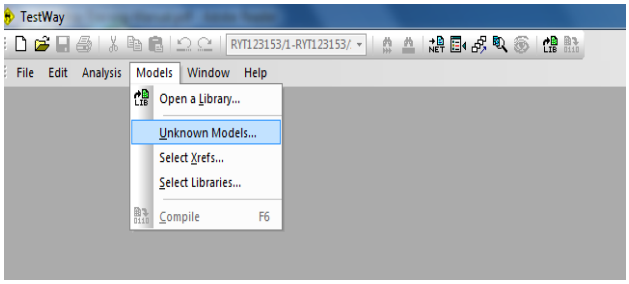


Figure 19. Project Name



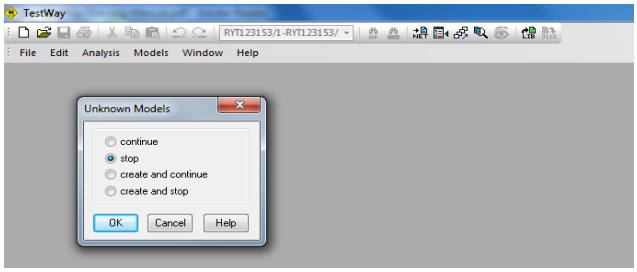


Figure 20. Unknown Models

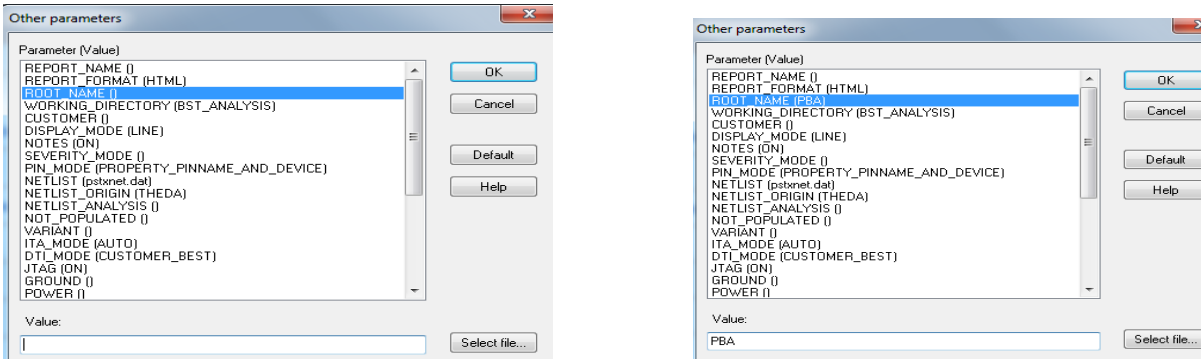


Figure 21. Root Name Configuration

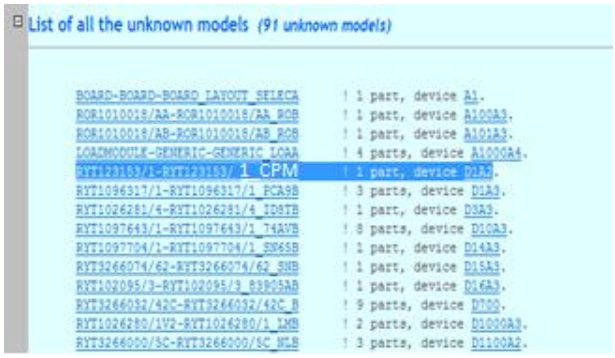


Figure 22. Unknown Models before BSCAN device Registration

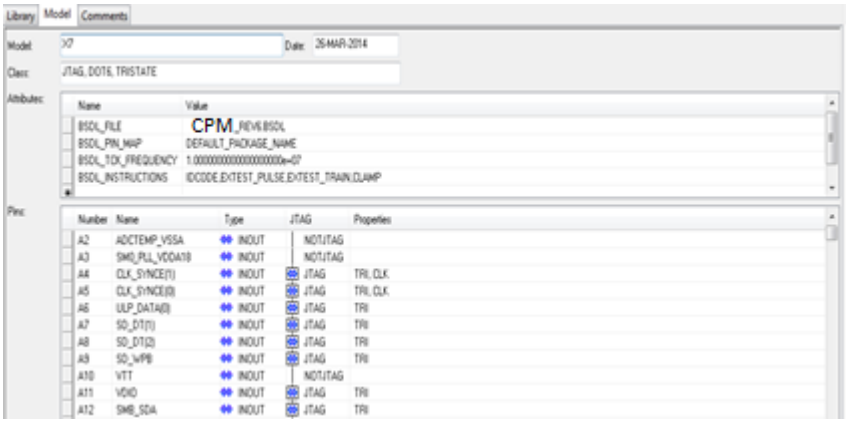


Figure 23. BSCAN Device details

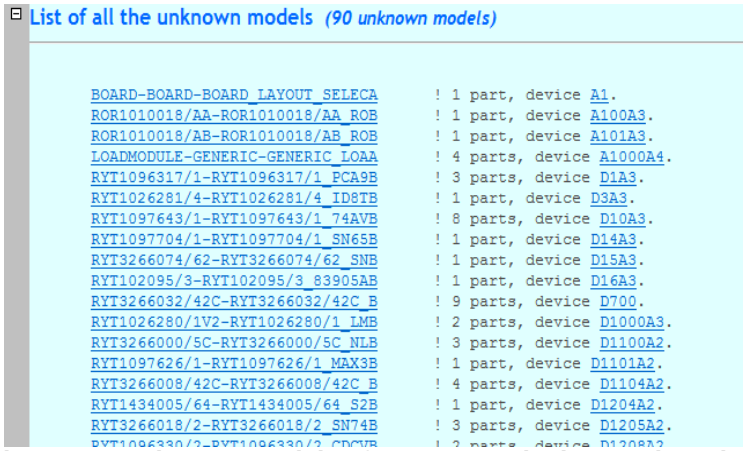


Figure 24. Unknown Models after BSCAN device Registration

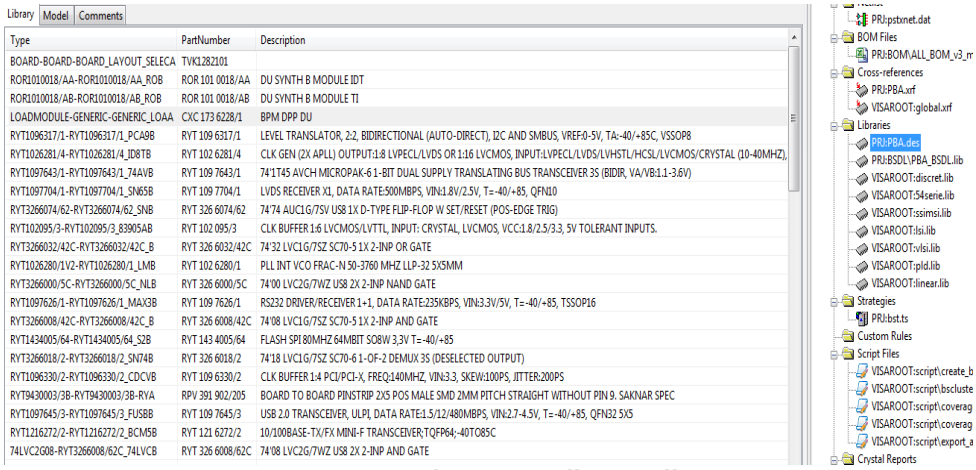


Figure 25. Library File

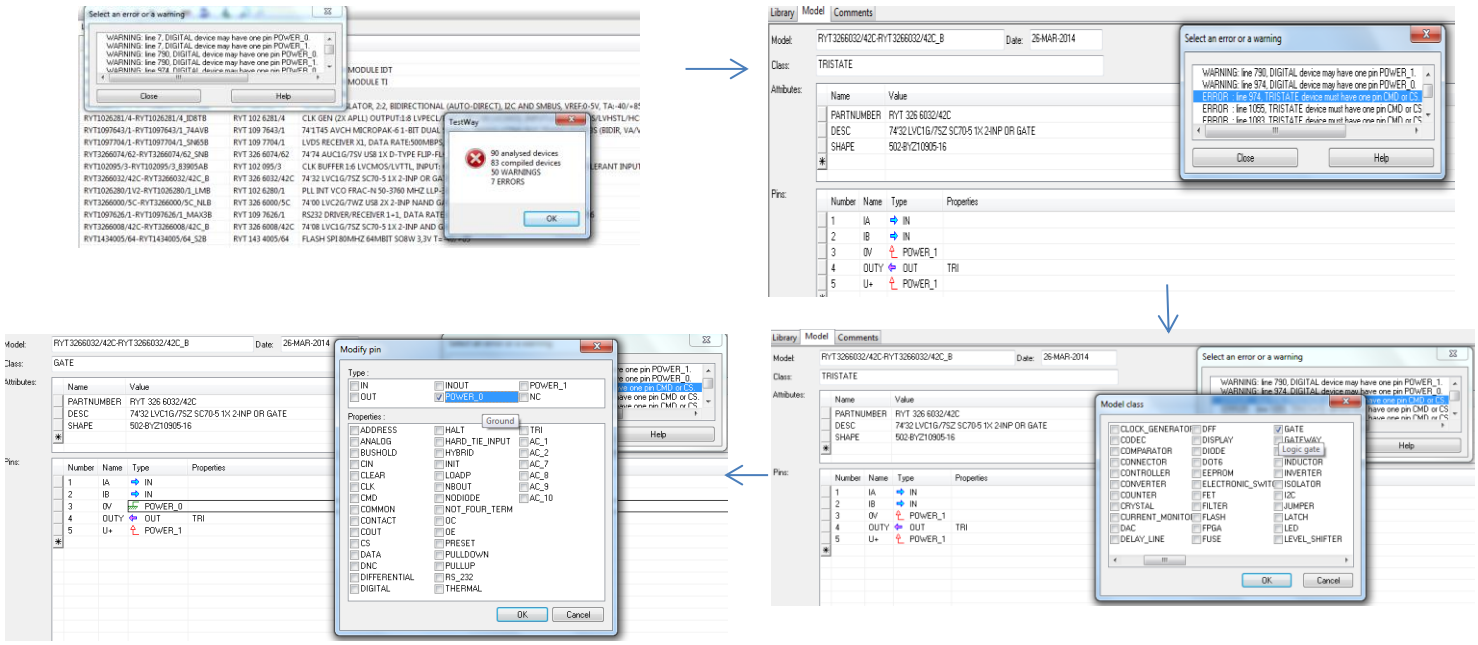


Figure 26. Modelling Phase

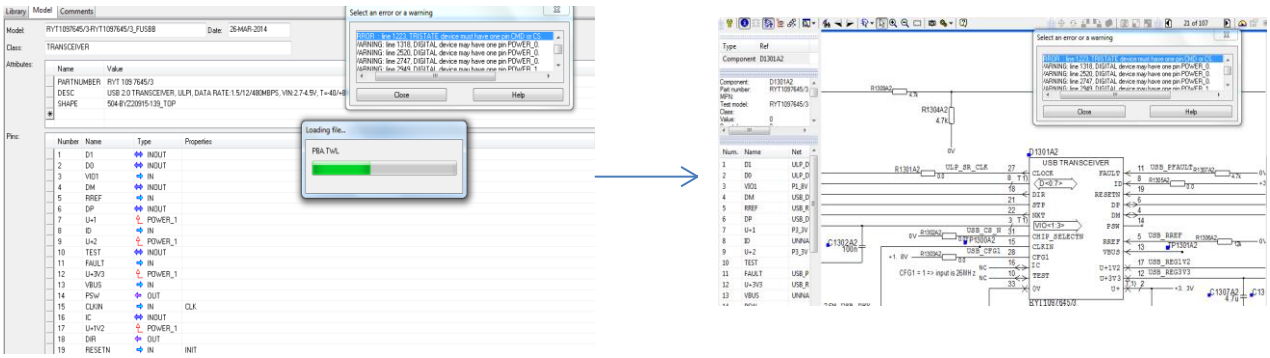


Figure 27. Library-QuadView Transition

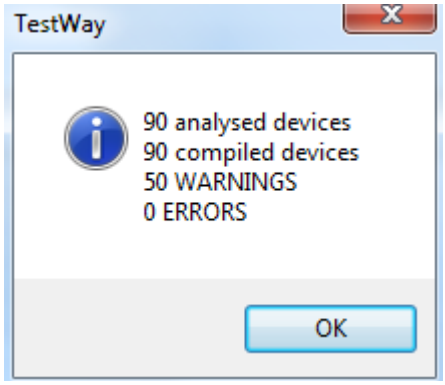


Figure 28. Library Compilation

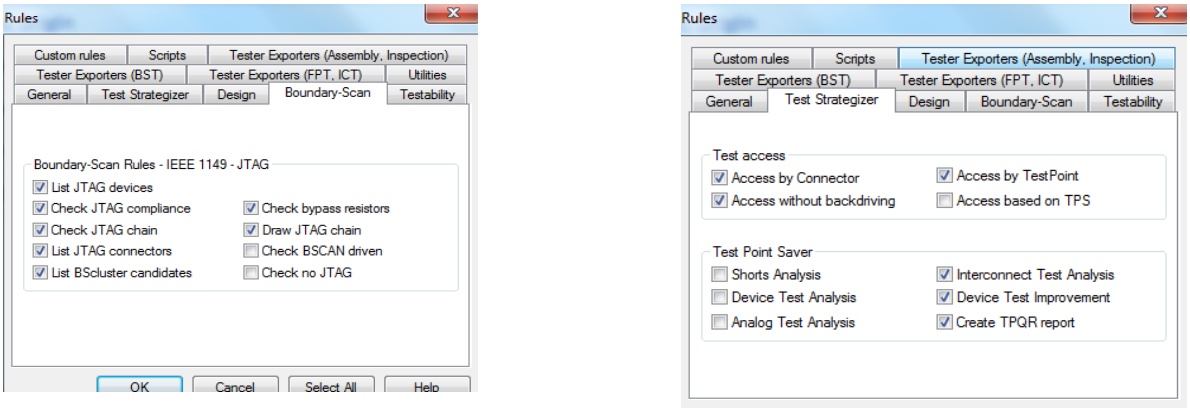


Figure 29. BSCAN Rules Configuration

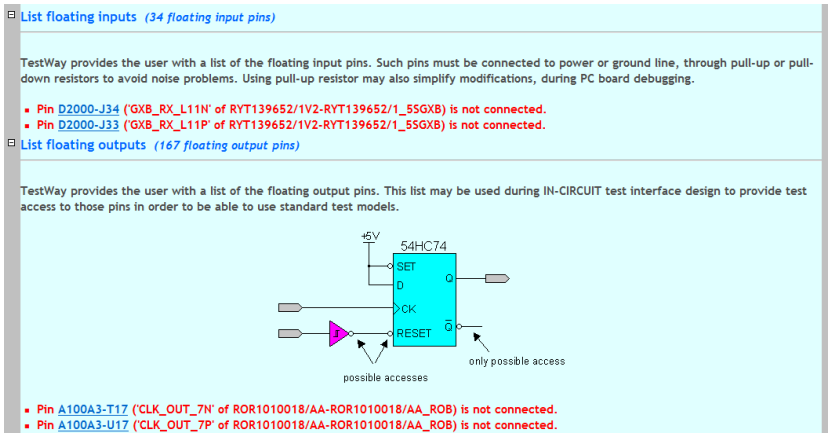


Figure 30. Floating Inputs

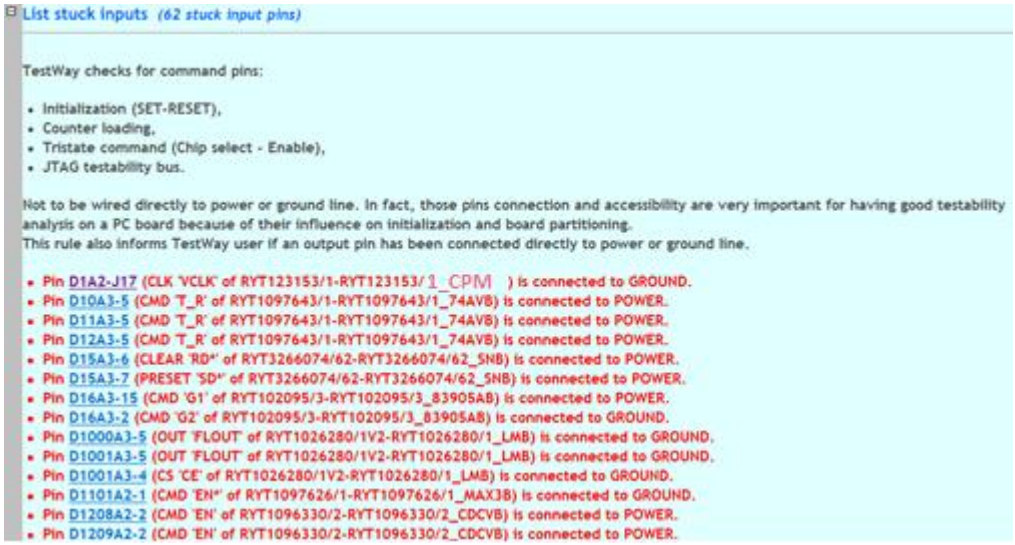


Figure 31. Stuck Inputs

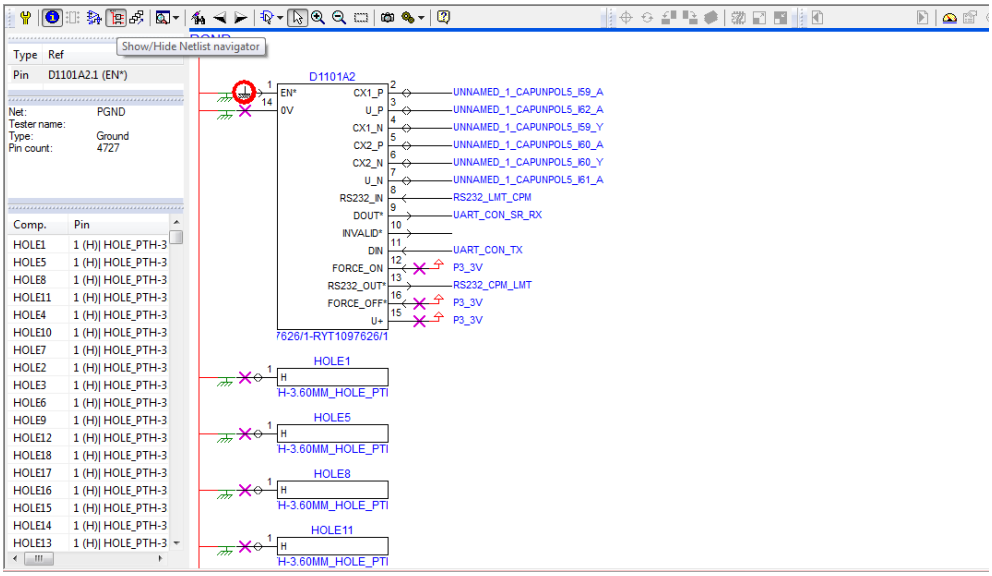


Figure 32. Stuck Input Pin

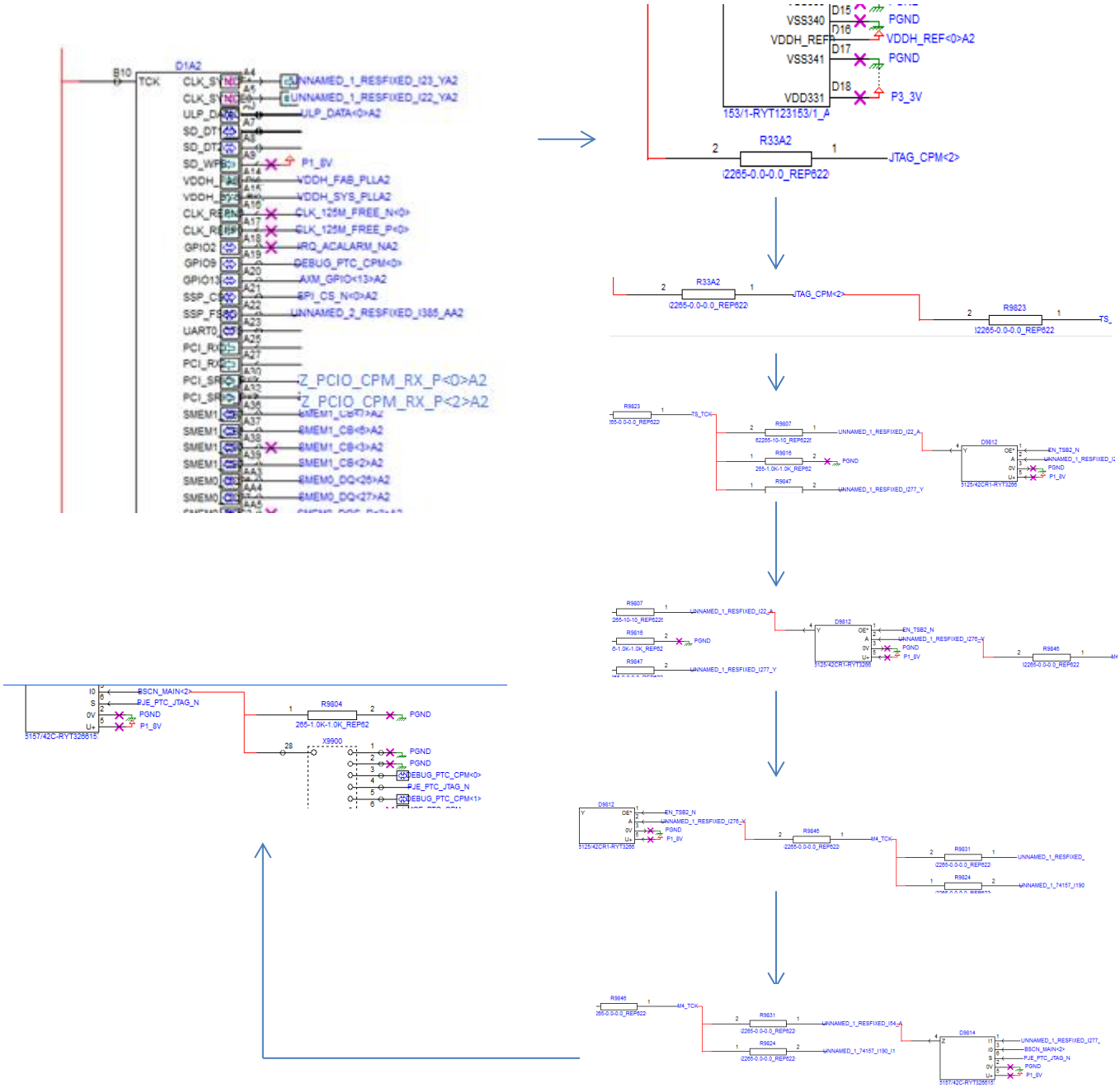


Figure 33. Check JTAG Chain

- BUFFER Device D1505A2 (RYT3262017/C-RYT3262017/C_SN74B) is not surrounded by JTAG cells.
- BUFFER Device D9801 (74LVC244-RYT3310244/3C_74LVC24B) is not surrounded by JTAG cells.
- BUFFER Device D9812 (RYT3266125/42CR1-RYT3266125/42B) is not surrounded by JTAG cells.
- BUFFER Device D9815 (RYT3266014/42C-RYT3266014/42C_B) is not surrounded by JTAG cells.
- BUFFER Device D3000A2 (RYT3266007/52C-RYT3266007/52C_B) is surrounded by 50.00% of JTAG cells.
- BUFFER Device D3001A2 (RYT3266007/52C-RYT3266007/52C_B) is surrounded by 50.00% of JTAG cells.
- RAM Device D2002A2 (RYT1441512/2-RYT1441512/2_MT41B) is surrounded by 100.00% of JTAG cells.
- RAM Device D2003 (RYT1441510/6-RYT1441510/6_MT41B) is surrounded by 100.00% of JTAG cells.
- RAM Device D2003A2 (RYT1441512/2-RYT1441512/2_MT41B) is surrounded by 100.00% of JTAG cells.
- RAM Device D2012A2 (RYT1441512/2-RYT1441512/2_MT41B) is surrounded by 100.00% of JTAG cells.
- RAM Device D2013A2 (RYT1441512/2-RYT1441512/2_MT41B) is surrounded by 100.00% of JTAG cells.
- RAM Device D3010A2 (RYT1441516/1-RYT1441516/1_MT41B) is surrounded by 96.49% of JTAG cells.
- RAM Device D3011A2 (RYT1441516/1-RYT1441516/1_MT41B) is surrounded by 96.49% of JTAG cells.
- RAM Device D3020A2 (RYT1441516/1-RYT1441516/1_MT41B) is surrounded by 96.49% of JTAG cells.
- RAM Device D3021A2 (RYT1441516/1-RYT1441516/1_MT41B) is surrounded by 96.49% of JTAG cells.
- RAM Device D3030A2 (RYT1441516/1-RYT1441516/1_MT41B) is surrounded by 96.49% of JTAG cells.
- RAM Device D3031A2 (RYT1441516/1-RYT1441516/1_MT41B) is surrounded by 96.49% of JTAG cells.
- RAM Device D3040A2 (RYT1441516/1-RYT1441516/1_MT41B) is surrounded by 96.49% of JTAG cells.
- RAM Device D3041A2 (RYT1441516/1-RYT1441516/1_MT41B) is surrounded by 96.49% of JTAG cells.

Figure 34. BSCAN Cluster

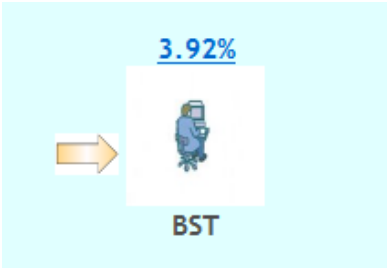


Figure 35. Basic BSCAN Coverage Value

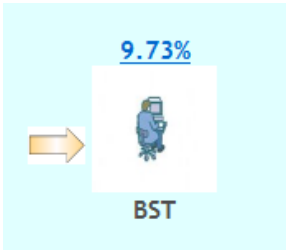


Figure 36. Enhanced BSCAN Coverage Value

Step 31
Add access to the following nets:
Node M2_TCK with pin D9810-4 ('1B1' of RYT1097643/1-RYT1097643/1_74AVB).
Node UNNAMED_1_RESFIXED_I82_A_1 with pin D9810-3 ('1A1' of RYT1097643/1-RYT1097643/1_74AVB).
This allows component testing of the following device: D9810.
Device D9810 (RYT1097643/1-RYT1097643/1_74AVB) is covered at 100.00%.
Cumulative totals:
51 nets with added access.
67 additional testable devices.

Figure 37. BSCAN Cluster D9810 Candidate

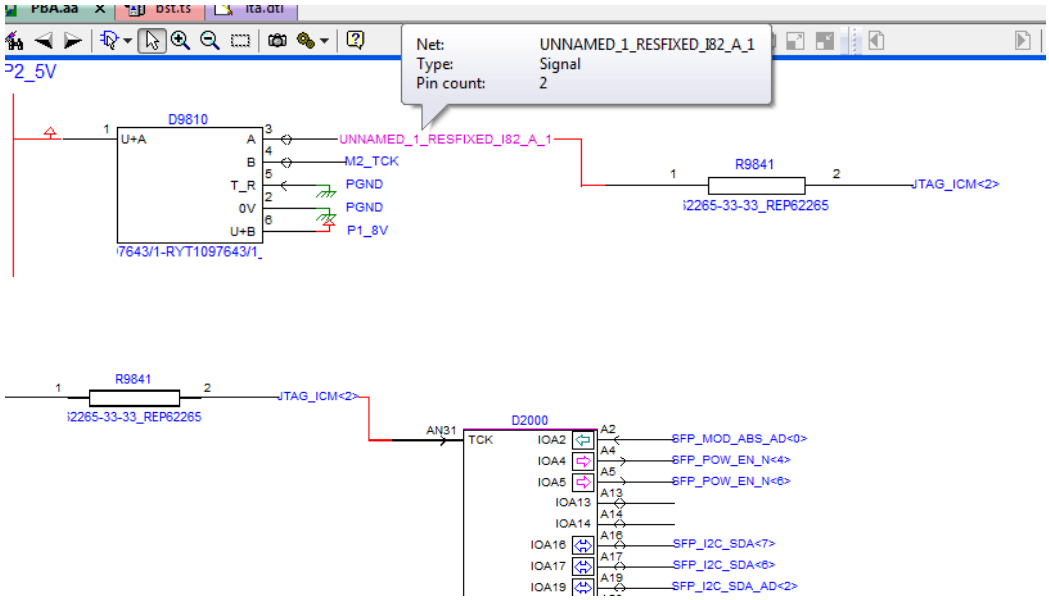


Figure 38. BSCAN Cluster D9810 Checking

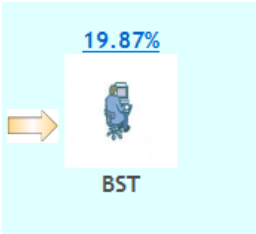


Figure 39. Final BSCAN Coverage Value

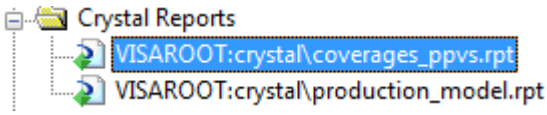


Figure 40. Coverage Report Generation Script File

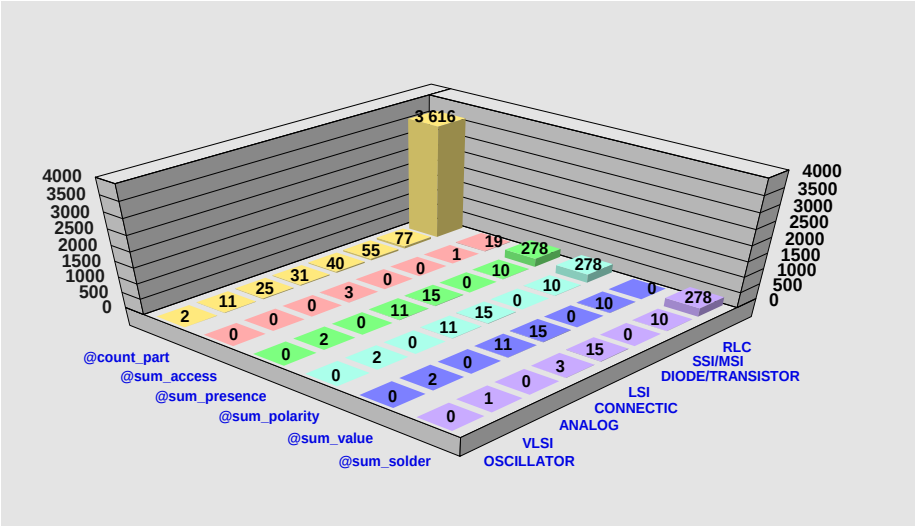


Figure 41.Coverages per defect category by component complexity

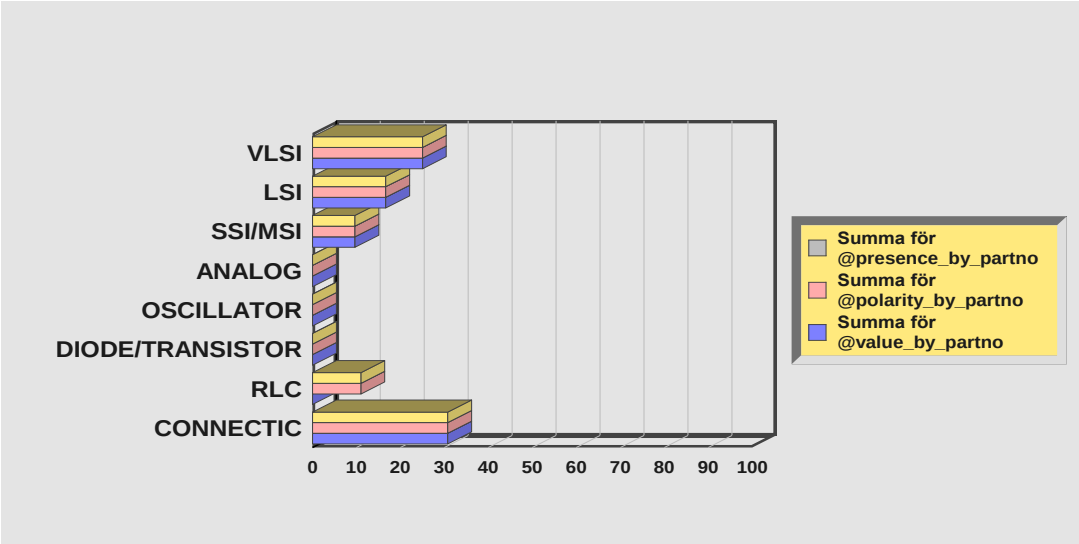


Figure 42.Partnumber coverage by component complexity

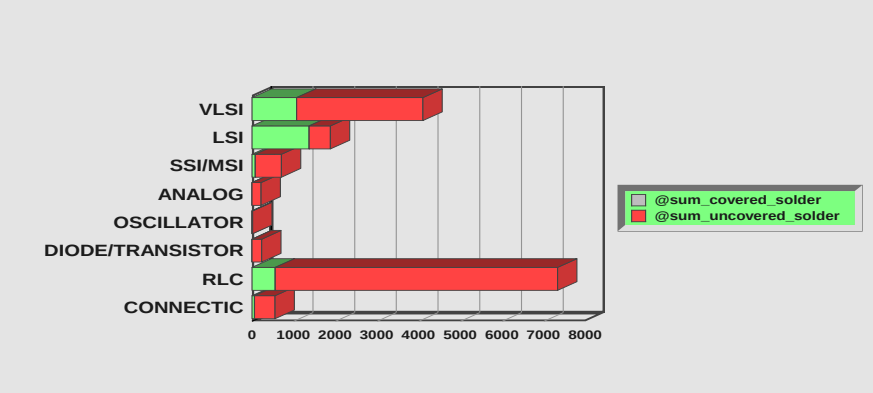


Figure 43. Soldering coverage by component complexity

TestWay - Coverage testability report — Board PBA

CONNECTIC parts					Coverage (%)						(ppm)
ref	type	function	shape	pin cnt	access	presen	polari	value	solder	weigh	uncov
X9800	RPV380250/002R1-RPV	CONNECTOR	512-BYZ20901-5	4	0,00	0,00	0,00	0,00	0,00	0,00	130
X2001	RPV380250/002R1-RPV	CONNECTOR	512-BYZ20901-5	4	50,00	100,00	100,00	100,00	50,00	84,62	20
RPV 380 250/002	Quantity of parts:		2		25,00	50,00	50,00	50,00	25,00	42,31	150
	Covered solders / total solders:		2 / 8								

Figure 44. Coverage Testability Report

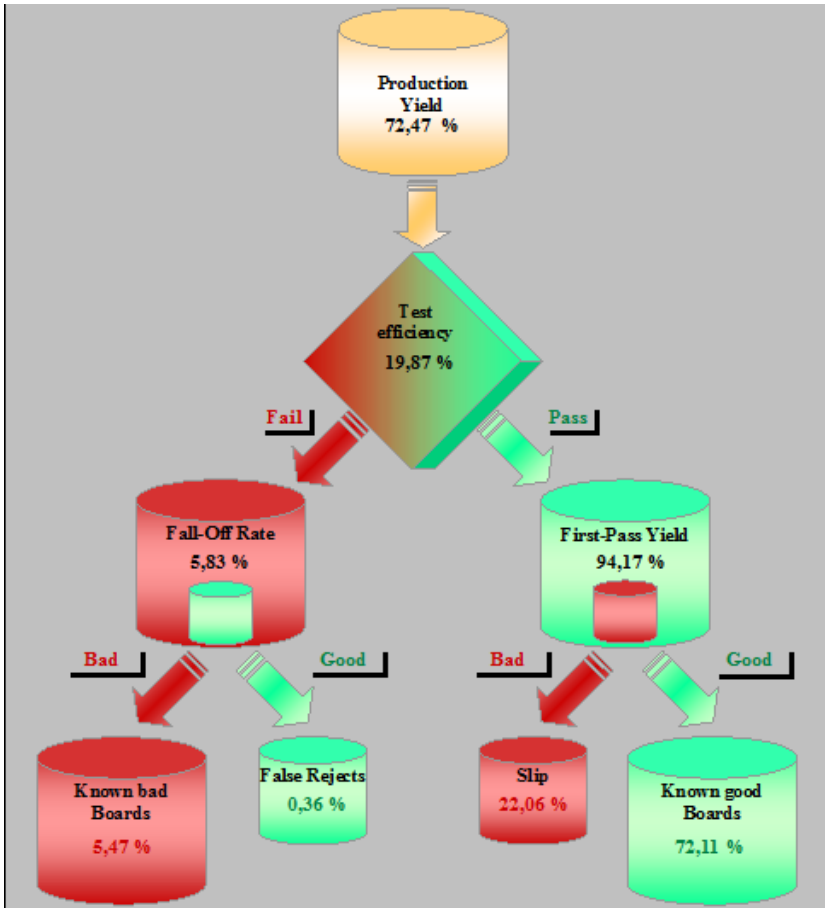


Figure 45. Production Yield Diagram

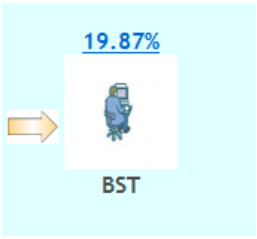


Figure 46. BSCAN Final Coverage Value

TEST REPORT FOR BST

BOARD NAME / NUMBER	PBA	DATA PROCESSING REPORT	WRK:PBA.htm
COMPONENT	4085	COMPONENT TESTED	8.79%
NET ACCESS	1.85%	BOARD SCORE	19.87%

DEVICE TYPE	TOTAL NUMBER (PARTS OR PINS)	Number of well tested	Number of partially tested	Number of not tested
Not identified	19 Parts	0.0% (0)	0.0% (0)	100.0% (19)
Integrated Circuit	206 Parts (7258 Pins)	15.0% (31)	1.0% (2)	84.0% (173)
Diode	2 Parts	0.0% (0)	0.0% (0)	100.0% (2)
Capacitor	2025 Parts	0.0% (0)	0.0% (0)	100.0% (2025)
Resistor	1441 Parts	20.5% (295)	0.0% (0)	79.5% (1146)
Inductor	144 Parts	0.0% (0)	0.0% (0)	100.0% (144)
Crystal	2 Parts	0.0% (0)	0.0% (0)	100.0% (2)
Switch	1 Parts	0.0% (0)	0.0% (0)	100.0% (1)
Connector	21 Parts (464 Pins)	0.0% (0)	52.4% (11)	47.6% (10)
Not Mounted	600 Parts			
Mechanical	5 Parts			
Total	3861 Parts (605 ignored)	8.4% (326)	0.3% (13)	91.2% (3522)

Figure 47. Test Report for BSCAN

TEST REPORT FOR BST

TESTABILITY REPORT FOR INTEGRATED CIRCUIT

REF. DES	PART NO.	PINS	COVERAGE SCORE	CONNECTION SCORE	COMMENT
A100A3	ROR1010018/AA-ROR1010018/AA_ROB	375	0.00%	0.00%	
A101A3	ROR1010018/AB-ROR1010018/AB_ROB	375	0.00%	0.00%	
D1000A3	RYT1026280/1V2-RYT1026280/1_LMB	33	0.00%	0.00%	
D1001A3	RYT1026280/1V2-RYT1026280/1_LMB	33	0.00%	0.00%	

Figure 48. Test Report for BSCAN-Integrated Circuits

D16A3	RYT102095/3-RYT102095/3_83905AB	16	0.00%	0.00%	
D1A2	RYT123153/1-RYT123153/1_CPM	1708	43.99%	43.74%	JTAG test
D1A3	RYT1096317/1-RYT1096317/1_PCA9B	8	0.00%	0.00%	
D2000	RYT139652/1V2-RYT139652/1_5SGXB	1152	28.47%	27.85%	JTAG test
D2000A4	RYT1266114/1-RYT1266114/1_LM50B	10	0.00%	0.00%	
D2002A2	RYT1441512/2-RYT1441512/2_MT41B	78	100.00%	100.00%	Cluster
D2003	RYT1441510/6-RYT1441510/6_MT41B	96	100.00%	100.00%	Cluster
D2003A2	RYT1441512/2-RYT1441512/2_MT41B	78	100.00%	100.00%	Cluster
D2006A3	RYT1097617/1-RYT1097617/1_CDCLB	17	0.00%	0.00%	
D2007A3	RYT1097617/1-RYT1097617/1_CDCLB	17	0.00%	0.00%	
D2012A2	RYT1441512/2-RYT1441512/2_MT41B	78	100.00%	100.00%	Cluster
D2013A2	RYT1441512/2-RYT1441512/2_MT41B	78	100.00%	100.00%	Cluster
D3000A2	RYT3266007/52C-RYT3266007/52C_B	5	100.00%	100.00%	Cluster
D3001A2	RYT3266007/52C-RYT3266007/52C_B	5	100.00%	100.00%	Cluster

Figure 49. Test Report for BSCAN-Tested BSCAN Cluster Components

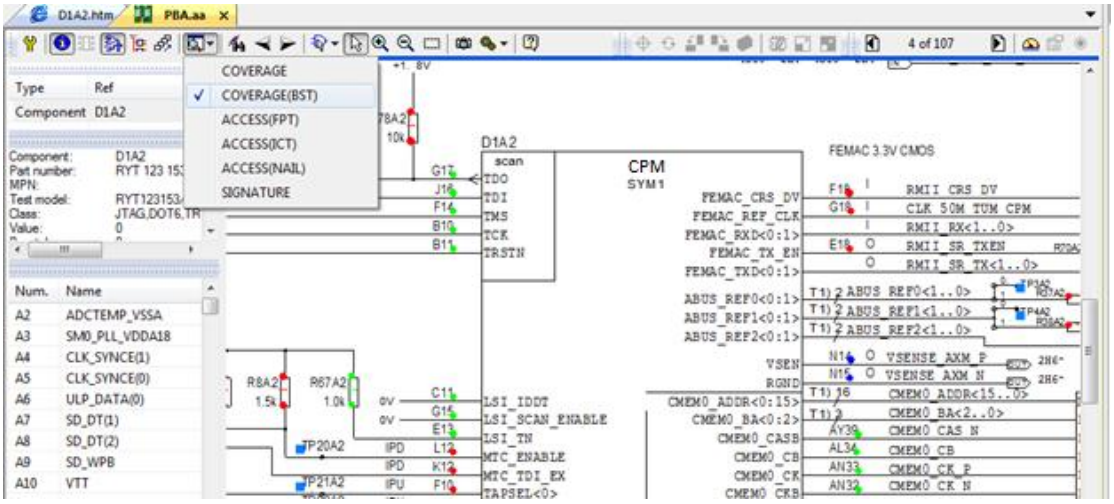


Figure 50. BSCAN Coverage Visualization

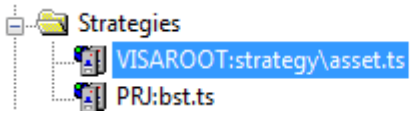


Figure 51. Loading ASSET test strategy file

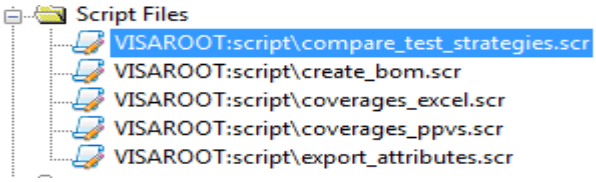


Figure 52. Loading Compare_Test_Strategies script file

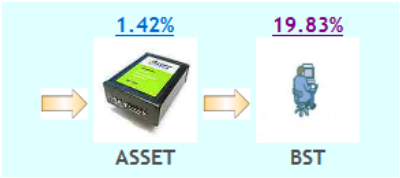


Figure 53. Comparing Real and Theoretical BSCAN Coverage Value

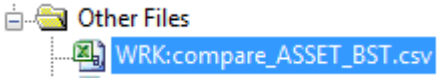


Figure 54. Compare_ASSET_BST .csv file

	ref	type	shape	function	ASSET	BST	coverage_presence_ASSET	coverage_presence_BST
12	D1A2	RYT123153/1-RYT123153/1_CPM	501-BYZ20920-522_TOP	DOT6			100.00	100.00
13	D2000	RYT139652/1V2-RYT139652/1_5SGXB	506-BYZ20920-241_TOP	DOT6			100.00	100.00
14	D2002A2	RYT1441512/2-RYT1441512/2_MT41B	506-BYZ20920-390_TOP	RAM		Cluster	0.00	100.00
15	D2003	RYT1441510/6-RYT1441510/6_MT41B	502-BYZ20920-385_TOP	RAM		Cluster	0.00	100.00
16	D2003A2	RYT1441512/2-RYT1441512/2_MT41B	506-BYZ20920-390_TOP	RAM		Cluster	0.00	100.00

Figure 55. Compare_ASSET_BST Report

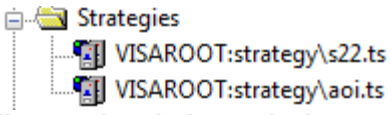


Figure 56. Loading Real and Theoretical AOI test strategy files

```
set test_setup("AOI","Inspection","{PreReflow,PostReflow}") = "PostReflow"
set test_setup("AOI","Side","{Top,Bottom,Both}") = "Both"
set test_setup("AOI","Mounting_Techno","{THT,SMD,Both") = "Both"
set test_setup("AOI","OCR","{Yes,No}") = "No"
```

Figure 57. Configuring AOI test strategy file

Figure 58. Theoretical VS Real AOI Coverage Value

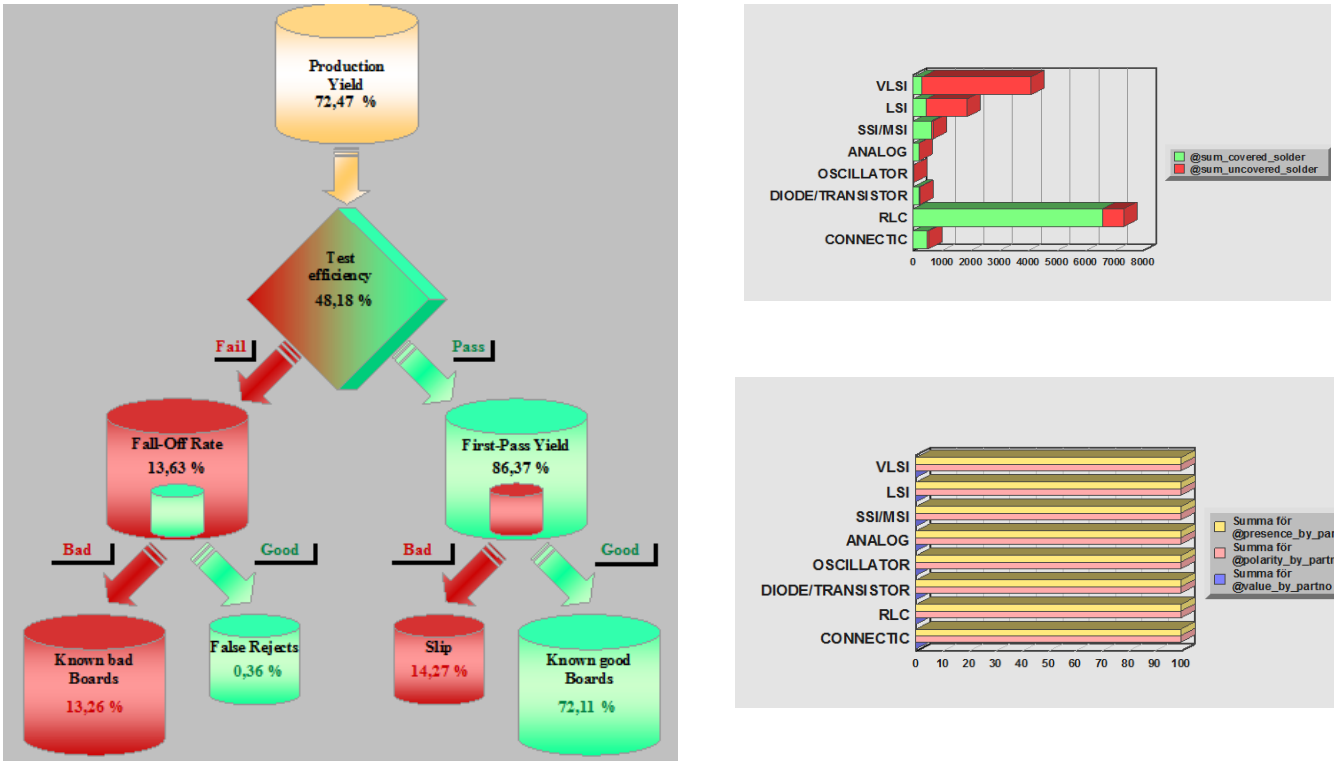


Figure 59. AOI Theoretical Coverage Report

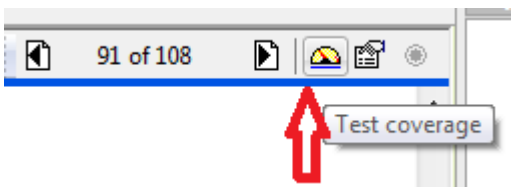


Figure 60. Capturing Primary Converter

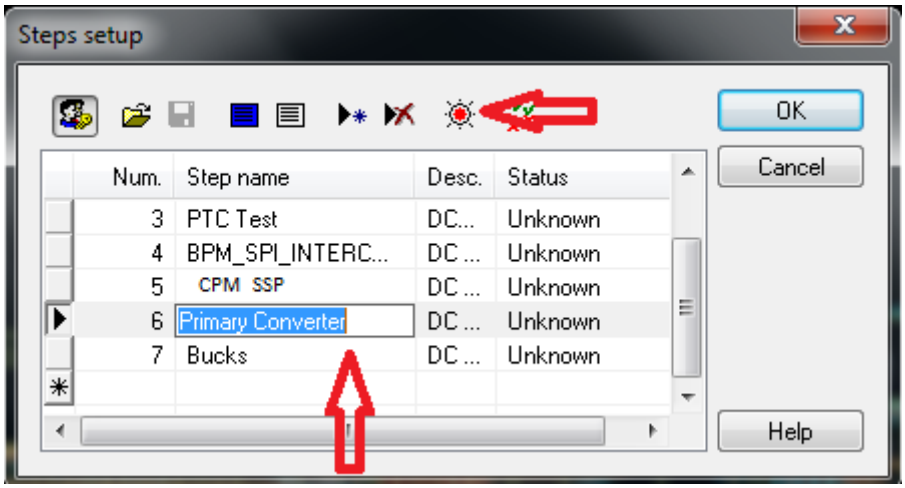


Figure 61. Primary Converter Test Step Creation

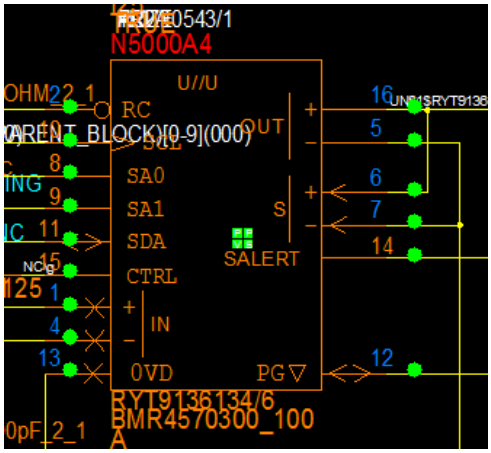


Figure 62. Covering Primary Converter for FT

Figure 63. Voltage Divider Covering

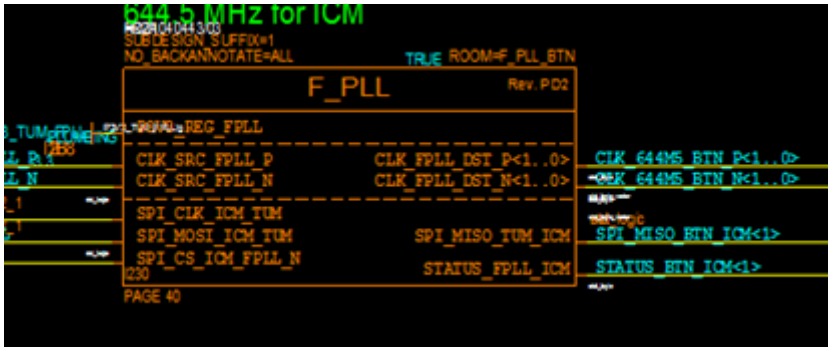


Figure 64. F_PLL Component

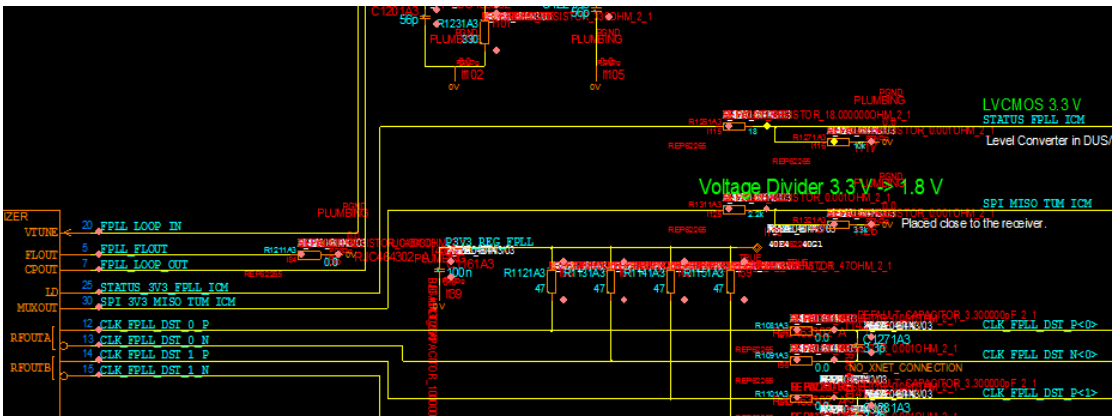


Figure 65. STATUS_FPLL_ICM Signal Covering

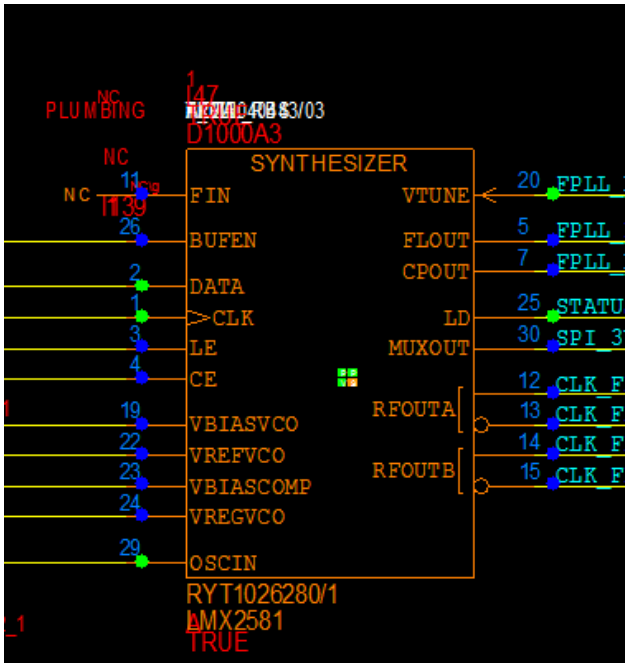


Figure 66. Covering Pins of the Synthesizer

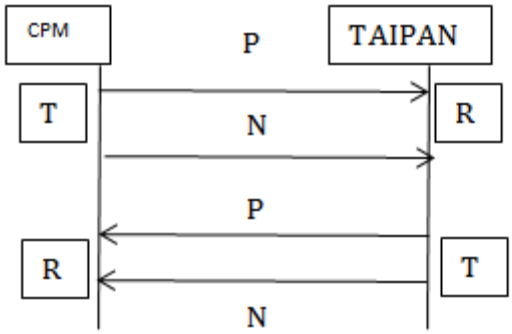


Figure 67. CPM-TAIPAN Connection

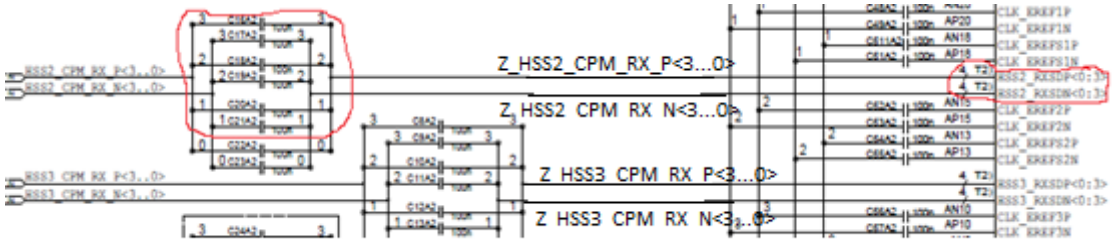


Figure 68. Covering Capacitors connected to the CPM Receivers

```
D1A2-AW9,1,100.0,0,0
D1A2-AY10,1,100.0,0,0
D1A2-AY12,1,100.0,0,0
D1A2-AY13,1,100.0,0,0
D1A2-AY14,1,100.0,0,0
```

Figure 69. Choosing which pins of the bus to ignore

Figure 70. TAIPAN Pins Covering

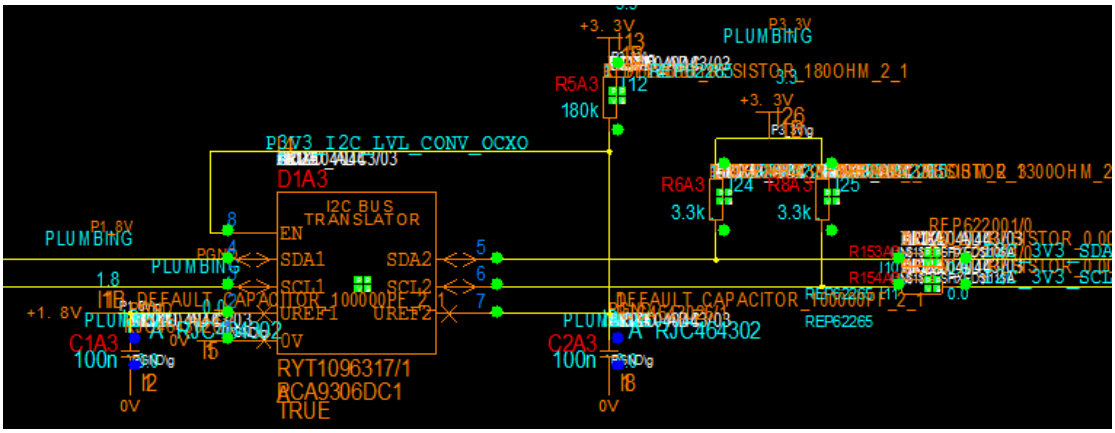


Figure 71. OCXO_ADC Block Covering

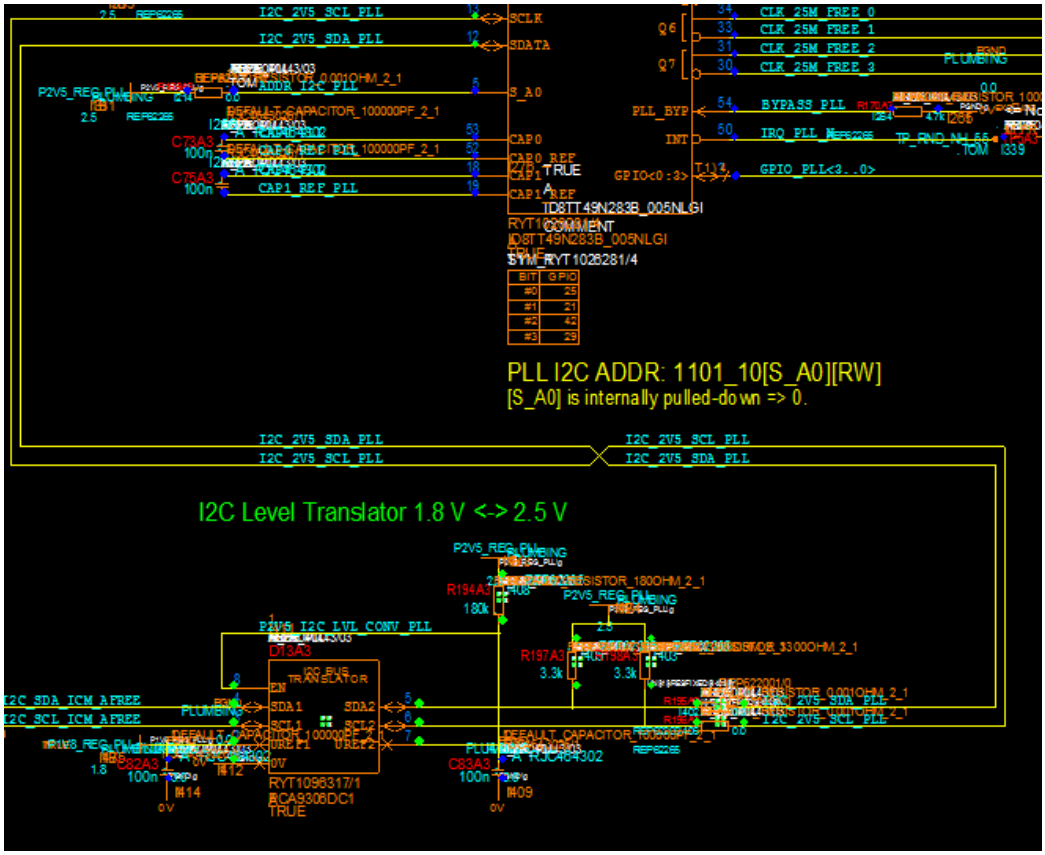


Figure 72. Covering I2C AFREE Signals

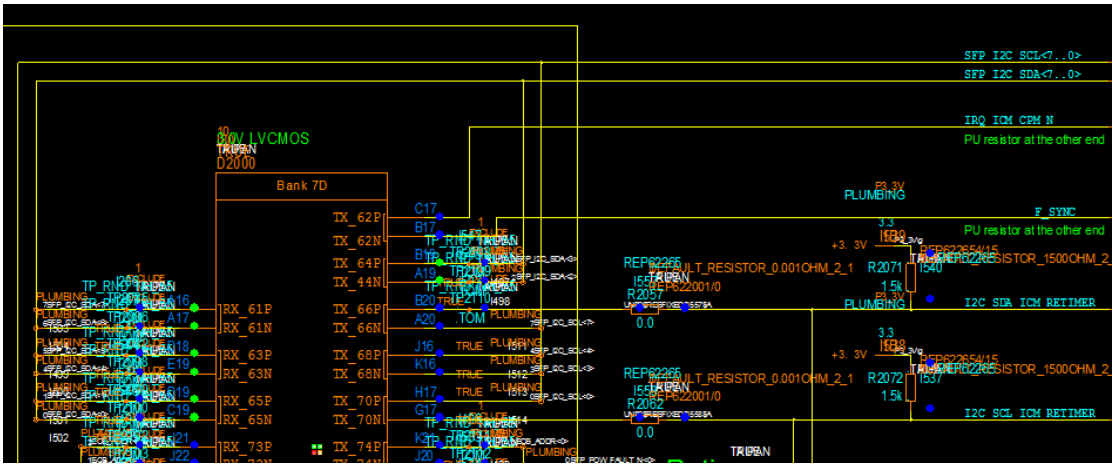


Figure 73. Covering 8 Pins of TAIPAN connected to 8 SFP connectors

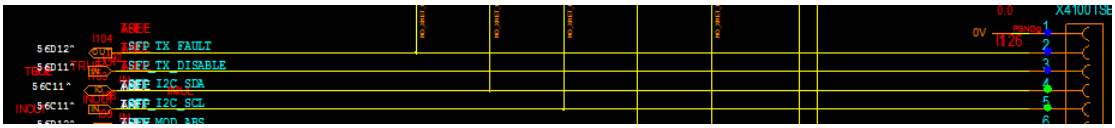


Figure 74. Covering Pins in SFP Connector

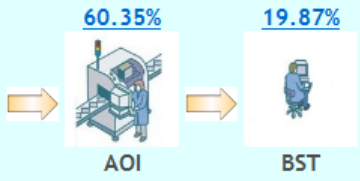
- Pin D1A2-P1 (SMEM0_DQ0) is equivalent to pin D1A2-Y41 (SMEM1_DQ0).
- Pin D1A2-H7 (SMEM0_RESETB) is equivalent to pin D1A2-AF35 (SMEM1_RESETB).
- Pin D1A2-N7 (SMEM0_PARERRB) is equivalent to pin D1A2-AA33 (SMEM1_PARERRB).
- Pin D1A2-P2 (SMEM0_DQ1) is equivalent to pin D1A2-Y40 (SMEM1_DQ1).
- Pin D1A2-R3 (SMEM0_DQ2) is equivalent to pin D1A2-W39 (SMEM1_DQ2).
- Pin D1A2-R4 (SMEM0_DQ3) is equivalent to pin D1A2-W38 (SMEM1_DQ3).
- Pin D1A2-P6 (SMEM0_DQ4) is equivalent to pin D1A2-Y36 (SMEM1_DQ4).
- Pin D1A2-P7 (SMEM0_DQ5) is equivalent to pin D1A2-Y35 (SMEM1_DQ5).

Figure 75. Module Matcher

567	MEM_BIST	MEM_BIST	2	D1A2	AJ41	SMEM1_RASB	-1.000000e+000	0.00	501-BYZ20920-522_TOP	D1A2	E1	SMEM0_RASB
568	MEM_BIST	MEM_BIST	2	D1A2	AC34	SMEM1_CASB	-1.000000e+000	0.00	501-BYZ20920-522_TOP	D1A2	K8	SMEM0_CASB
569	MEM_BIST	MEM_BIST	2	D1A2	AB35	SMEM1_FB_IN	-1.000000e+000	0.00	501-BYZ20920-522_TOP	D1A2	L8	SMEM0_FB_IN
570	MEM_BIST	MEM_BIST	2	D1A2	AB33	SMEM1_FB_OUT	-1.000000e+000	0.00	501-BYZ20920-522_TOP	D1A2	M8	SMEM0_FB_OUT
571	MEM_BIST	MEM_BIST	2	D1A2	AD34	SMEM1_WEB	-1.000000e+000	0.00	501-BYZ20920-522_TOP	D1A2	L9	SMEM0_WEB

Figure 76. Equivalence between two components

Create TPQR report



- The coverage report file is WRK:combined\index.htm.

Figure 80. Combining BST-AOI Coverage Value

"D5000" Component scores

Presence	Polarity	Value	Solder
0.00%	0.00%	0.00%	0.00%

Pin name	Node	Tests	Connection score	Comment
1	PGND		-1.00%	
2	REF_DISABLE_ICM_PHY		0.00%	
3	UNNAMED_2_RESFIXED_I293_Y		0.00%	
4	UNNAMED_2_RESFIXED_I294_A		0.00%	
5	D5000_POWER		-1.00%	
6	UNNAMED_2_CAPUNPOL5_I301_Y		0.00%	
7	UNNAMED_2_CAPUNPOL5_I302_Y		0.00%	
8			-1.00%	
9	CLK_125M_SB1S0P		0.00%	
10	CLK_125M_SB1S0N		0.00%	
11	CLK_125M_SB1S1P		0.00%	
12	CLK_125M_SB1S1N		0.00%	
13	CLK_125M_SB1S2P		0.00%	
14	CLK_125M_SB1S2N		0.00%	
15			0.00%	
16			0.00%	
17	PGND		-1.00%	

Figure 81.AOI ignoring PGND pins

"D5000" COMPONENT SCORES

PRESENCE	POLARITY	VALUE	SOLDER
100.00%	100.00%	0.00%	90.00%

PIN NAME	NODE	TESTS	CONNECTION SCORE	COMMENT
1	PGND		90.00%	
2	REF_DISABLE_ICM_PHY		90.00%	
3	UNNAMED_2_RESFIXED_I293_Y		90.00%	
4	UNNAMED_2_RESFIXED_I294_A		90.00%	
5	D5000_POWER		90.00%	
6	UNNAMED_2_CAPUNPOL5_I301_Y		90.00%	
7	UNNAMED_2_CAPUNPOL5_I302_Y		90.00%	
8			90.00%	
9	CLK_125M_SB1S0P		90.00%	
10	CLK_125M_SB1S0N		90.00%	
11	CLK_125M_SB1S1P		90.00%	
12	CLK_125M_SB1S1N		90.00%	
13	CLK_125M_SB1S2P		90.00%	
14	CLK_125M_SB1S2N		90.00%	
15			90.00%	
16			90.00%	
17	PGND		90.00%	

Figure 82.AOI covering PGND pins

TEST REPORT FOR AOI,BST

BOARD NAME / NUMBER	PBA	DATA PROCESSING REPORT	WRK:PBA.htm
COMPONENT	4085	COMPONENT TESTED	100.00%
NET ACCESS	1.85%	BOARD SCORE	75.27%

Figure 773. Combined AOI and BST Coverage Value

Create TPQR report

The coverage report file is WRK:combined\index.htm.

Figure 84. Combining FT-AOI-BST Coverage

TEST REPORT FOR FT,AOI,BST

BOARD NAME / NUMBER	PBA	DATA PROCESSING REPORT	WRK:PBA.htm
COMPONENT	4085	COMPONENT TESTED	100.00%
NET ACCESS	1.85%	BOARD SCORE	77.00%

Figure 85. Combined FT-AOI-BST Coverage

References

[1] Elena Dubrova, “Fault-Tolerant Design”, Springer, 2013

[2] “Board DfT & Test Coverage Analyzer”, http://www.aster-technologies.com/pub/en-TestWay-Brochure.pdf last accessed: 2014-05-06

[3] Peter Collins, Christophe Lotz, Dominique Wiatrowski, “Functional Board Test – Coverage Analysis: what does it mean when a functional test passes “, http://www.aster-technologies.com/pub/en-EBTW06%20Paper_1.2.pdf last accessed: 2014-05-06

[4] Christophe Lotz,”DfX- Defect Universe and Test Coverage”, Test Forum 2013 Nov 26-27 Tallin, Estonia

[5] ”An introduction to Automatic Optical Inspection”, DCB Automation, http://www.testsolve.co.uk/uploads/Document_Downloads/Introduction%20to%20Automatic%20Optical%20Inspection.pdf last accessed: 2014-05-06

[6] “Fast Generation of HW Test&Programming applications”, JTAG Technologies, 2013, http://www.jtag.com/en/system/files/downloads/JTAG%20ProVision%20Datasheet_0.pdf last accessed: 2014-05-06

[7] “Automated optical inspection”, Wikipedia, http://en.wikipedia.org/wiki/Automated_optical_inspection last accessed: 2014-05-06

[8] “Boundary Scan Tutorial”, Corelis, http://www.corelis.com/education/Boundary-Scan_Tutorial.htm last accessed: 2014-05-06

[9] “Boundary Scan”, http://en.wikipedia.org/wiki/Boundary_scan last accessed: 2014-05-06

[10] “13. IEEE 1149.1 (JTAG) Boundary-ScanTesting for MAX II Devices”, Altera, http://www.altera.com/literature/hb/max2/max2_mii51014.pdf last accessed: 2014-05-06

[11] “Did you know this about Boundary Scan?” Agilent Technologies, http://www.home.agilent.com/owc_discussions/thread.jspa?messageID=102758 last accessed: 2014-05-06

[12] “Functional Test: A Final Manufacturing Step”, Nexlogic, http://www.nexlogic.com/services/pcb-testing/functional-testing.aspx last accessed: 2014-05-06

[13] Stig Oresjo, “How defect coverage as a variable can be used to determine test and inspection strategies”, Agilent Technologies, BTW 2005

[14] Adam W Ley, “Defect Coverage for Non-Intrusive Board Tests”, ASSET InterTech, 2009

[15] “Integrated Boundary Scan Solutions”, Acculogic, http://www.electronics.pcb.com.pl/electronics/downloads/acculogic003.pdf last accessed: 2014-05-06

[16] “ScanWorks Boundary Scan Test Development”, ASSET Intertech, https://www.asset-intertech.com/Media/en-US/Documents/Boundary-Scan-Test/Product-Brief/BST_Development_Product_Brief.pdf last accessed: 2014-05-06

[17] “SIMATIC IT Unicam Test Epert”, Siemens, http://www.automation.siemens.com/mcms/mes/en/mescomponents/npiforelectronics/Documents/brochure_simaticit_unicam_testexpert_en.pdf last accessed: 2014-05-06

[18] “Functional Test Coverage Assessment Project Board Test Workshop”, iNEMI, http://www.docstoc.com/docs/42733733/Functional-Test-Coverage-Assessment-Project-Board-Test-Workshop last accessed: 2014-05-06

[19] Tony Tailor, “iNEMI Statement of Work”, iNEMI, 2007

[20] http://mantech-sj.com/technical_papers/AOIAdvantages.pdf last accessed: 2014-03-13

[21] ”Reflow Soldering”,http://en.wikipedia.org/wiki/Reflow_soldering last accessed: 2014-05-06

[22] “AOI / AXI technology in the latest development in SMT”, http://www.tyh-ele.com/index.aspx?menuid=26&type=articleinfo&lanmuid=50&infoid=101&language=en last accessed: 2014-05-06

[23] Robert H.Seemer, “Defects Per Million Opportunities (DPMO)”, 2010

[24] “Design For Test (DFT) Guidelines for Boundary-Scan Testing”, Corelis, http://www.corelis.com/whitepapers/Boundary-Scan_DFT_Guidelines.pdf last accessed: 2014-05-06

[25] “SCAN Express DFT Analyzer”, http://www.flashtech.com.sg/Catalogue/ScanExpress_DFT_Analyzer%5B1%5D.pdf last accessed 2014-05-27

[26] Teradyne D2B Strategist Datasheet, http://www.electronicsdatasheets.com/pdf-datasheets/teradyne/d2b-strategist/ last accessed 2014-05-27

TRITA ICT-EX-2014:42