



**POLITECNICO
DI TORINO**



AUTOMATIC TOOL INTEGRATION INTO AN INDUSTRIAL ELECTRONIC BOARD PRODUCTION HARDWARE FAULT COVERAGE ANALYSIS PROCESS

Luca Ledda

Computer Engineering
Embedded Systems
POLITECNICO DI TORINO

SUPERVISORS:

Prof. Matteo Sonza Reorda

Test Eng. Paolo Caruzzo

Test Eng. Roberto Saroglia

Test Eng. Paolo Nigro

Turin, October 2018

ABSTRACT

In Magneti Marelli S.p.A. Powertrain very high quality is essential. To achieve such a high quality, the production test must be capable of detecting all potential defects introduced during the production process.

The adopted test line involves ICT (In-Circuit Test, Bed of Nails), FT (Functional Test) and AOI (Automated Optical Inspection) strategies. The quality of the test line, which is an indicator of how many defects the line is capable of detecting, is resumed into the coverage document, manually produced by the Magneti Marelli test engineers through the use of personal reasoning and know-how.

The purpose of this project is to integrate the hardware fault coverage process with a customized tool in order to obtain as much as possible an automatic flow to generate the final test coverage document aiming to break down, where possible, timing and human error barriers.

OVERVIEW

The thesis is organized as follows:

In the **INTRODUCTION** part, a brief history of the Magneti Marelli S.p.A. is reported ([chapter 1](#)). After that, the focus is moved to the Powertrain testing team ([chapter 2](#)) such that highlighting its responsibilities and interactions within the company hierarchy.

In the **PRODUCTION TEST** part, an overview of the Magneti Marelli Powertrain production line is reported focusing on all those defects that the final product can be affected at the end of manufacturing ([chapter 3](#)). Then, the test strategies in the flow impacting on the hardware fault coverage are described ([chapter 4](#)), highlighting for each technique general theoretical concepts in form of equipment involved, potentialities, limits and possible future improvements.

In the **HARDWARE FAULT COVERAGE** part, a first introduction to the test coverage theory is presented ([chapter 5](#)), reporting traditional metric aspects and coverage document contents. Secondly, the Magneti Marelli Powertrain test coverage is described ([chapter 6](#)), highlighting characteristics of their proprietary metric and coverage document. Proprietary coverage advantages and disadvantages are then reported in order to define the company motivations to migrate towards an automatic flow.

In the **COVERAGE AUTOMATIC TOOL** part, TestWay Express is firstly introduced ([chapter 7](#)) in order to secondly describe the feasibility analysis performed by Magneti Marelli ([chapter 8](#)) aiming to understand whether the tool was suitable for the intended purposes. Once the results of the analysis have been documented, a third chapter ([chapter 9](#)) describes the customization introduced to the tool to best achieve the goals previously stated by the company.

The **CONCLUSION** part includes the difficulties encountered during the whole experience and the adopted solution to cope with them, a panoramic of the achieved, missed goals and some possible directions that the project can take in future ([chapter 10](#)).

In the **APPENDIX** part, a User Manual is reported to instruct a Test Engineer on the use of the customized tool such that approaching the Hardware Fault Coverage flow in an automatic way ([Appendix A](#)).

CONTENTS

I	INTRODUCTION	1
1	MAGNETI MARELLI HISTORY	2
1.1	The past	2
1.2	Nowadays	3
2	POWERTRAIN TESTING TEAM	5
2.1	MM Powertrain Product	5
2.2	Testing Team Responsibility	6
2.3	Test Engineer Interactions	9
II	PRODUCTION TEST	11
3	PRODUCTION LINE	12
3.1	From PCB to Final Product	12
3.1.1	Front End	12
3.1.2	Back End	17
3.1.3	Prototypes and Validation Phases	20
3.2	Potential Defects	21
3.2.1	Soldering Defects	22
3.2.2	Component Defects	24
3.2.3	Defects Summary	25
4	TEST STRATEGY LINE	27
4.1	AOI	28
4.1.1	AOI Equipment	28
4.1.2	AOI Techniques	30
4.1.3	AOI Limits	31
4.1.4	AOI Detection Capabilities	31
4.1.5	AOI Future	33
4.2	ICT	34
4.2.1	ICT Equipment	34
4.2.2	ICT Techniques	37
4.2.3	ICT Limits	49
4.2.4	ICT Detection Capabilities	52
4.2.5	ICT Future	54
4.3	EOL	56
4.3.1	EOL Equipment	56
4.3.2	EOL Techniques	58
4.3.3	EOL Limits	61
4.3.4	EOL Detection Capabilities	63
4.3.5	EOL Future	64

III	HARDWARE FAULT COVERAGE	66
5	TEST COVERAGE	67
5.1	Coverage Introduction	67
5.2	Coverage Metrics	68
5.2.1	MPS	69
5.2.2	PPVS	69
5.2.3	PCOLA/SOQ	70
5.2.4	Conclusion	72
5.3	Defect Universe	73
5.4	Coverage Document	75
5.5	Final Test Coverage	77
6	MM-PWT TEST COVERAGE	79
6.1	Proprietary Metric	79
6.2	Proprietary Coverage Document	82
6.2.1	Creation	82
6.2.2	Compilation	86
6.2.3	Coverage Computation	88
6.3	Advantages and Disadvantages	89
6.4	Goals	89
IV	COVERAGE AUTOMATIC TOOL	90
7	TESTWAY EXPRESS	91
7.1	Overview	91
7.2	Features	92
7.2.1	Input Board Data	92
7.2.2	Modeling the Components	93
7.2.3	Place the Probes	94
7.2.4	Coverage Analysis	95
8	TWE FEASIBILITY ANALYSIS	96
8.1	Case of Study	96
8.2	Analysis	97
8.3	Results	99
9	TWE CUSTOMIZATION	100
9.1	MM-PWT Customized Environment	100
9.1.1	BoardView.ini	101
9.1.2	visalib.ini	102
9.2	Creation of a New Project	103
9.3	Files Import	105
9.3.1	BOM File Import	105
9.3.2	CAD File Import	107
9.4	Component Classification	108
9.4.1	Component Classifier	109
9.4.2	Local Classification Storage	111
9.4.3	Global Classification Storage	111

9.4.4	Models	113
9.5	Probe Analyzer	115
9.5.1	Layout for Test Validation	116
9.5.2	Design for Test Validation	121
9.6	Test Coverage	124
9.6.1	Preventive Coverage Analysis	125
9.6.2	Real Coverage Analysis	130
9.6.3	Declaration Method Coverage Analysis	130
9.7	Coverage Reports	133
9.7.1	MM-PWT TWE Scripts	134
9.7.2	Graphical Report	138
9.7.3	Numerical Report	140
V	CONCLUSION	141
10	CONCLUSIONS	142
10.1	Achieved and Missed Goals	142
10.1.1	Achieved Goals	142
10.1.2	Missed Goals	142
10.2	Future of the Automatic Flow in MM-PWT	144
VI	APPENDIX	145
A	USER MANUAL	146
A.1	How To	146
A.1.1	Integrate the Customized Environment	147
A.1.2	Create New Project	148
A.1.3	Import Files	149
A.1.4	Classify the Components	151
A.1.5	Launch Probe Analyzer (DfT Checker)	155
A.1.6	Generate Preventive Coverage Analysis	158
A.1.7	Generate ICT Uncovered Pins List	162
A.1.8	Generate Real Coverage Analysis	164
A.1.9	Generate Final Coverage	166
	BIBLIOGRAPHY	167
	RINGRAZIAMENTI	170

LIST OF FIGURES

Figure 1	MM-PWT product - Engine Management System	5
Figure 2	TTE - Position in the MM-PWT hierarchy	6
Figure 3	Example - Electronically Controlled Throttle body	7
Figure 4	TTE - Team structure	8
Figure 5	MM-PWT Production Line - Front-End	13
Figure 6	Production Process - Solder Paste Stenciling	14
Figure 7	Production Process - Pick and Place Machine	15
Figure 8	Production Process - Reflow Oven	15
Figure 9	Production Process - THT Component	16
Figure 10	Production Process - THT Solder Geometry [1]	17
Figure 11	MM-PWT Production Line - Back-End	18
Figure 12	Infant Mortality - Bathtub Curve [37]	19
Figure 13	MM-PWT Production Line - Prototype Versions	20
Figure 14	PCBA Defect - Solder Bridging	22
Figure 15	PCBA Defect - Solder Excess	22
Figure 16	PCBA Defect - Solder Poor	23
Figure 17	PCBA Defect - Solder Balling	23
Figure 18	PCBA Defect - Solder Void	23
Figure 19	PCBA Defect - Lifted Component	24
Figure 20	PCBA Defect - Shifted Component	24
Figure 21	IC Device - Polarization Mark	25
Figure 22	PCBA Defects - BGA Soldering	26
Figure 23	MM-PWT - Test Strategy Line	27
Figure 24	AOI - Equipment	28
Figure 25	AOI - Camera and Lights cooperation [38]	29
Figure 26	AOI - Lifted Component detection [7]	33
Figure 27	AXI - BGA defects detection [15]	33
Figure 28	ICT - Equipment	34
Figure 29	ICT - Bed Of Nails [16]	35
Figure 30	ICT - ATE-Bed Of Nails communication system	36
Figure 31	ICT - "Growing Power" flow	38
Figure 32	ICT - Test Sequences Flow	38
Figure 33	ICT Technique - Floating Test	39
Figure 34	ICT Technique - Link Test	40
Figure 35	ICT Technique - Short Test	40
Figure 36	ICT Technique - Analog Guarding Technique	41
Figure 37	ICT Technique - 2-wires Resistor Test	42
Figure 38	ICT Technique - 4-wires Resistor Test (Kelvin Technique)	43
Figure 39	ICT Technique - Capacitor Test	44

Figure 40	ICT Technique - Inductor Test	44
Figure 41	ICT Technique - Diode Test	45
Figure 42	ICT Technique - Diode Scan Test	46
Figure 43	ICT Technique - Transistor Test	47
Figure 44	ICT Technique - Operational Amplifier Test	47
Figure 45	ICT Technique - Digital Test	48
Figure 46	ICT Limit - Filter Capacitor	49
Figure 47	ICT Limit - RC Parallel	50
Figure 48	ICT Limit - Clamping Diodes Parallel	51
Figure 49	ICT Limit - Tied IC Pins	51
Figure 50	ICT Limit - Shorted NC pins	52
Figure 51	EOL - Test Firmware	57
Figure 52	EOL - Architecture Capability	58
Figure 53	EOL - TestStand Layer	59
Figure 54	EOL - DUT_OFF Test	60
Figure 55	EOL - ATE Instruments/Panel Impedance	62
Figure 56	EOL - Safety Tolerance Ranges	63
Figure 57	Test Coverage Flow	68
Figure 58	Test Coverage - MPS and PPVS metrics	70
Figure 59	Defect Universe - Far from reality	74
Figure 60	Defect Universe - Close to reality	75
Figure 61	Test Coverage Document - Flow	75
Figure 62	Coverage Document Compilation - Circuit Samples	77
Figure 63	Transistor (3-pins)	79
Figure 64	MM-PWT _{coverage} - Coverage Document Creation Process	82
Figure 65	MM-PWT _{coverage} Document Compilation - Reference Circuit	86
Figure 66	Production Flow - Waterfall Approach (without a Tool)	91
Figure 67	Production Flow - V-Shape Approach (with a Tool)	92
Figure 68	TWE Features - Hardware Fault Coverage flow	92
Figure 69	TWE Input Board Data - Added Value	93
Figure 70	TWE Feasibility Analysis - Flow	97
Figure 71	TWE Customization - MARELLI Folder Structure	100
Figure 72	TWE Customization - New Project Environment Structure	103
Figure 73	TWE Customization - New Project Flow-Links	104
Figure 74	TWE Customization - BOM Import	106
Figure 75	TWE Customization - Customized CAD Import	108
Figure 76	Component Classification - TWE Management	109
Figure 77	TWE Customization - Customized MASTERLIB Management	113
Figure 78	TWE Customization - Local and Global Models	114
Figure 79	Probe Analyzer - Concept	115
Figure 80	Probe Analyzer - Concept	116
Figure 81	MM-PWT Manufacturing Rules - Minimum TP Diameter	117
Figure 82	MM-PWT Manufacturing Rules - Minimum TPs Distance	117
Figure 83	MM-PWT Manufacturing Rules - Component Body Clearance	118

Figure 84	MM-PWT Manufacturing Rules - Tooling Hole Clearance . . .	118
Figure 85	MM-PWT Manufacturing Rules - Board Outline Clearance . . .	119
Figure 86	Probe Analyzer - Customized Profile (Layout Checker)	120
Figure 87	Probe Analyzer - Design Validation Concept	122
Figure 88	Probe Analyzer - Customized Profile (Design Checker)	123
Figure 89	TWE - Preventive Coverage Analysis Flow	126
Figure 90	TWE - Customized Preventive Coverage Analysis Flow	127
Figure 91	TWE - Customized Declaration Method Coverage Analysis Flow	131
Figure 92	TWE - Customized Coverage Report Creation Flow	133
Figure 93	TWE Customization - Graphical Report Creation Flow	139
Figure 94	TWE - The whole Coverage Analysis Flow	143
Figure 95	Hardware Fault Coverage - User Manual Flow	146
Figure 96	TWE MM-PWT Customized Environment - Main Window	147
Figure 97	TWE Environment - Layout View Interaction	150
Figure 98	Component Classifier - Missing Classification on Top	151
Figure 99	Component Classifier - Missing Models	152
Figure 100	Component Classifier - Generate Model File	152
Figure 101	Component Classifier - Edit Model File	152
Figure 102	Component Classifier - Add Information Model	153
Figure 103	Component Classifier - Add Models from Model Library	153
Figure 104	Component Classifier - Update MasterLib	154
Figure 105	Probe Analyzer - Main Window	155
Figure 106	Probe Analyzer - Layout Tab, Accessibility Information	156
Figure 107	Probe Analyzer - Configuration Summary	156
Figure 108	Probe Analyzer - Inaccessible Nets List	157
Figure 109	Test Program Quality - Component Classification Indicator	158
Figure 110	Test Program Quality - Deselect Test Strategy	159
Figure 111	Test Program Quality - Refresh Report Alert	159
Figure 112	Test Program Quality - View report	160
Figure 113	Test Program Quality - Board Coverage Info	160
Figure 114	Test Program Quality - Component Coverage Info	161
Figure 115	Test Program Quality - Pin Coverage Info	161
Figure 116	DfT Checker - Main Window	162
Figure 117	Preventive ICT - Uncovered Pins List	163
Figure 118	Final Coverage - Best Test Line	166

LIST OF TABLES

Table 1	Potential Defects - Solders Summary	25
Table 2	Potential Defects - Components Summary	26
Table 3	Potential Defects - BGA Summary	26
Table 4	AOI - Defects Detection Capabilities	32
Table 5	ICT Technique - Digital Test patterns	48
Table 6	ICT Technique - Digital Guarding Test patterns	49
Table 7	ICT - Defects Detection Capabilities	53
Table 8	EOL - Defects Detection Capabilities	63
Table 9	PCOLA/SOQ - Connection Properties	72
Table 10	Metrics - MPS, PPVS and PCOLA/SOQ relations	72
Table 11	Coverage Document Compilation - Document Example	76
Table 12	Test Coverage Percentage - Computation Example	78
Table 13	MM-PWT _{metric} - Adaptivity	80
Table 14	MM-PWT _{metric} - <i>FaultModel.ini</i> structure	81
Table 15	MM-PWT _{coverage} - Empty Document (Details Tab)	84
Table 16	MM-PWT _{coverage} - Empty Document (Summary Tab)	85
Table 17	MM-PWT _{coverage} - Compiled Document (Details Tab)	88
Table 18	TWE Feasibility Analysis - Mapping MM-PWT _{metric} to PPVS	98
Table 19	TWE Feasibility Analysis - Comparison Document	99
Table 20	BOM FUJI - File Format	105
Table 21	BOM TWE-compatible - File Format	105
Table 22	MM-PWT Manufacturing Rules - Minimum TP Diameter	117
Table 23	MM-PWT Manufacturing Rules - Minimum TPs Distance	117
Table 24	MM-PWT Manufacturing Rules - Component Body Clearance	118
Table 25	TWE Customization - MM-PWT Test Line Coverage Analysis	125
Table 26	TWE Customization - Declaration Method Step File	131
Table 27	TWE Customization - Declaration Method Part File	132
Table 28	TWE Customization - Declaration Method Pin File	132

LISTINGS

Listing 1	TWE Customization - BoardView.ini (CONFIGURATION) . .	101
Listing 2	TWE Customization - BoardView.ini (Component Classifier)	102
Listing 3	TWE Customization - visalib.ini	102
Listing 4	Files Import - BOM Grammar File	106
Listing 5	TWE Customization - Link BOM Import to Project	106
Listing 6	Component Data - CADENCE ALLEGRO export	107
Listing 7	Vias Data - CADENCE ALLEGRO export	107
Listing 8	TWE Customization - Link CAD Import to Project	108
Listing 9	TWE Customization - <i>BoardView.ini</i> (Classifier)	110
Listing 10	TWE Customization - MasterLib Grammar File	112
Listing 11	TWE Customization - Header MasterLib File	112
Listing 12	TWE Customization - Link MASTERLIB to the Project	112
Listing 13	TWE Customization - Link Models to the Project	113
Listing 14	TWE Customization - Missing Models Rule	115
Listing 15	TWE Customization - Link Test Line Script to Project	115
Listing 16	TWE Customization - Layout Checker Profiles (PA)	119
Listing 17	TWE Customization - Layout Checker Priority Rules (PA)	120
Listing 18	TWE Customization - Manufacturing Constraints	121
Listing 19	TWE Customization - Design Checker Profiles (PA)	123
Listing 20	TWE Customization - Design Checker Priority Rules (PA)	124
Listing 21	TWE Customization - Design Constraints	124
Listing 22	TWE Customization - Preventive ICT Default Settings	127
Listing 23	TWE Customization - Resistor Test Preventive ICT	128
Listing 24	TWE Customization - <i>Open</i> and <i>Short</i> Rules Preventive ICT	129
Listing 25	TWE Customization - <i>Presence</i> Rule Preventive ICT	129
Listing 26	TWE Customization - <i>Correct</i> Rule Preventive ICT	129
Listing 27	TWE Customization - <i>Live</i> Rule Preventive ICT	130
Listing 28	TWE Customization - FCT Test Program Variables	132
Listing 29	TWE Customization - FCT Component Variables	133
Listing 30	TWE Customization - FCT Pin Variables	133
Listing 31	TWE Customization - <i>MM_PWT_test_line.scr</i> file	134
Listing 32	TWE Customization - Link Test Line Script to Project	135
Listing 33	TWE Customization - <i>MM_PWT_coverage_ICT.scr</i> file	135
Listing 34	TWE Customization - Link Coverage Scripts to Project	136
Listing 35	TWE Customization - <i>print_uncoveredPins_ICT.scr</i> file	136
Listing 36	TWE Customization - <i>Component with uncovered Pin</i> Rule	137
Listing 37	TWE Customization - <i>Uncovered Pin</i> Rule	137
Listing 38	TWE Customization - Link Uncovered Pins ICT Scripts to Project	138

Listing 39	TWE Customization - Graphical Report Variables	140
Listing 40	TWE Customization - EOL Declaration Report Syntax	165

ACRONYMS

AOI Automated Optical Inspection

ATE Automatic Test Equipment

AXI Automated X-ray Inspection

BGA Ball Grid Array

BEV Battery Electric Vehicle

BOM Bill Of Material

DfT Design for Testability

DU Defect Universe

DUT Device Under Test

DUT Design Validation

ECU Electronic Control Unit

EEPROM Electrically Erasable Programming Read Only Memory

EOL End Of Line

FT Functional Test

HEV Hybrid Electric Vehicle

IC Integrated Circuit

ICE Internal Combustion Engine

ICT In-Circuit Test

MCU Micro Controller Unit

MDA Manufacturing Defects Analyzer

MM Magneti Marelli S.p.A.

MM-PWT Magneti Marelli Powertrain

MPS Material Placement Solder

MTTF Mean Time To Failure

NC	Not Connected
OA	Operational Amplifier
PA	Probe Analyzer
PCB	Printed Circuit Board
PCBA	Printed Circuit Board Assembled
PCOLA	Presence Correct Orientation Live Alignment
SOQ	Short Open Quality
PC	Personal Computer
PPVS	Presence Polarity Value Solder
PRI	Power Run-In
PV	Process Validation
R&D	Research & Development
SMD	Surface Mount Device
SMT	Surface Mount Technology
SPI	Solder Paste Inspection
SW	Software
TCU	Transmission Control Unit
THT	Through Hole Technology
TP	Test Point
TPGM	Test Program EOL
TTE	Testing and Tool Engineering
TWE	TestWay Express Tool

Part I

INTRODUCTION

MAGNETI MARELLI HISTORY

The Magneti Marelli S.p.A. history [10], from the origins (section 1.1) up to nowadays (section 1.2), is presented in this chapter.

1.1 The past

In 1891, Ercole Marelli founded the company bearing his name, specialised in the production of electrical devices and engines. After about 25 years, Fiat and the "Società anonima Ercole Marelli", in order to satisfy the growing mobility sector demand, established a partnership to start a mass production of magnets for internal combustion engine.

Magneti Marelli was founded on 8th October 1919 in equal parts by Fiat and "Società anonima Ercole Marelli". Their activities, from 1920, became a huge variety so that a in-house personnel training was launched to prepare employees on the study and design of new products related to automobiles, motorbikes, industrial motors, aviation and racing engines.

An important contribute of Magneti Marelli was related to the production of professional radio equipment for land, aeronautical and naval communications, in 1930. Five years later, a partnership with Bosch was established for the marketing of electrical equipment for automobiles and motorbikes.

Thanks to the acquired popularity, Magneti Marelli was chosen to test, design and build the first television broadcasts for RAI, from the camera to the TV, in 1939. In 1940, became also a supplier to the Italian Army, Navy and Air Force, for which it produced electrical equipment and mobile reception/transmission systems.

An important breakthrough was in the 1960s, when the Magneti Marelli activities gradually started to focus on the automotive industry. In fact, in 1963, the entire telecommunication sector was sold in order to acquire a company specialized in the manufacture of ignition sparkplugs but also to inaugurate the first automotive production facility in Brazil, in 1978 (one of the main hubs of the current company's activities). The study and production of electronic control devices for ignition and supply systems was definitively set up after the foundation of Marelli Autronica (joint venture between Magneti Marelli, Fiat and Weber).

In 1980s, the entire sector of automotive batteries and accumulators was completely controlled by Magneti Marelli which, thanks to the acquisition of some other companies, became a leader in the field of lighting, in-vehicle electronics and fuel control/supply systems.

The process of focusing only on the automotive field fully completed definitively in the 1990s, when strategies were adopted to strengthen the automotive core business and expanded it on the international markets. In the 1996, in fact, Magneti Marelli started its operation in China with the Guangzhou plant and some additional production facilities in Shanghai.

1.2 Nowadays

With the start of the new Millennium, the company expanded its vocation of major components manufacturer being able to design and produce complete automotive systems for leading carmakers.

In 1998, it started its activities in the area of satellite navigation. Between 2000 and 2001, Fiat decides to sell some of their business branches such as electronic systems and air-conditioning. The same year, Magneti Marelli took full control of Automotive Lighting, a joint venture with Bosch.

An important contribute to the environmental sustainability, safety and "intelligent" automobiles was given by Magneti Marelli in 2003, when it launched, in Brazil, a combustion engines able to run both on ethanol, gasoline or any blend of the two.

In 2009, after 90 years from its foundation, Magneti Marelli, helped by Fiat and Confindustria, extended its perimeter in the whole India and Shanghai. In the same years, it also developed the KERS, a system used in Formula I for the energy recovery under breaking.

Nowadays, Magneti Marelli counts 43'000 employees, 86 production centers and 14 R&D centers [11]. It is extended in 19 countries to offers international services through the following business lines:

- Automotive Lighting
- **POWERTRAIN**
- Electronic Systems
- Suspension Systems
- Exhaust Systems
- Plastic Components and Modules
- After Market Parts and Services
- Motorsport

A particular attention is related to the "Powertrain" (PWT) business line, on which the thesis is focused on.

The [chapter 2](#) introduces the Powertrain line and describes all the aspects concerning the team in charge of testing PWT products.

POWERTRAIN TESTING TEAM

This chapter involves an overview description of the Magneti Marelli Powertrain product (section 2.1), the responsibility of the Magneti Marelli team in charge of testing it (section 2.2) and the interactions of the Test Engineer figure towards other teams within the company hierarchy (section 2.3).

2.1 MM Powertrain Product

Magneti Marelli Powertrain is a business line dedicated to engines and transmissions production for cars, motorbikes and light vehicles. In other words, "Powertrain" refers to *"those systems that assist generation and delivering of power to the surface on which the vehicle is operating"* [31] (in the case of automotive industry, the road is the surface under analysis).

The concept behind Magneti Marelli Powertrain products is to provide an Engine Management System which interacts between driver and motor. That system senses the driver intents in order to properly actuate the motor engine such that user requests are satisfied in terms of power provided to the vehicle (electric power, engine torque and exhaust gas).

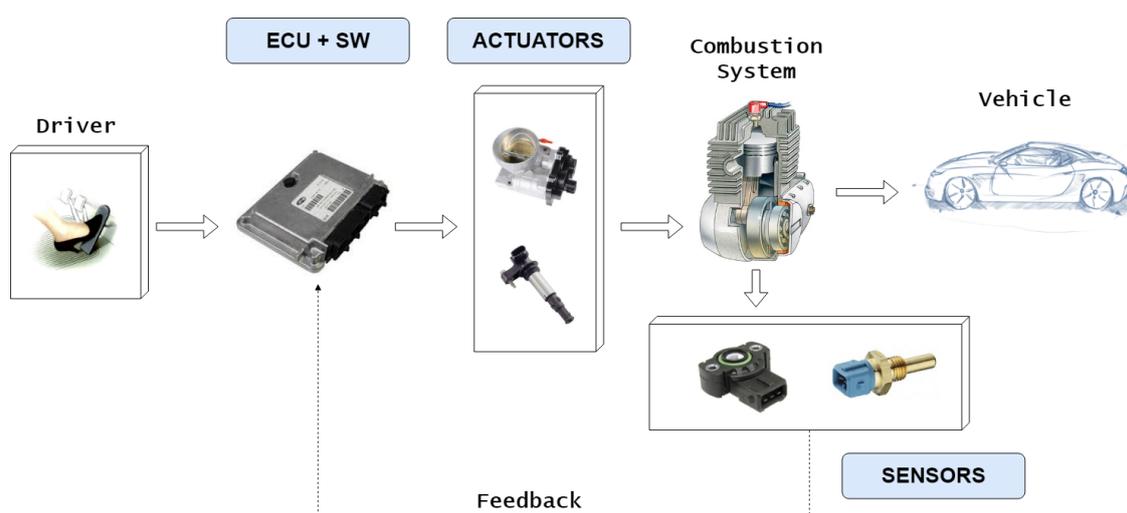


Figure 1: MM-PWT product – Engine Management System

The [Figure 1](#) is a draft of the previously introduced flow. From left to right, the ECU, which runs a firmware, detects the acceleration driver intent. According to this, it actuates the combustion engine which, in turn, converts combustion to powertrain for the vehicle. It is important to notice that sensors help the ECU in this process, precisely to understand what is happening in the combustion engine during actuation.

This process works thanks to the cooperation of several components. In particular, MM-PWT produces the parts of the mechanism which is composed, as the blue shapes in the [Figure 1](#) highlight, by the ECU/TCU (Electronic/Transmission Control Unit) running a software, the **sensors** and the **actuators**. This production refers to the market of transmissions, gasoline and diesel (ICE - Internal Combustion Engine) engine control. However, with the growth of electric vehicles, hybrids and alternative fuels, there is a much more diverse mix of fuel types on the roads, and as a result, also engine controls for HEV (Hybrid Electric Vehicle) and BEV (Battery Electric Vehicle) have become objects of MM-PWT production market [6].

2.2 Testing Team Responsibility

In the [section 2.1](#) it has been explained what MM-PWT business line produces: the ECU+SW, sensors and actuators (composing the Powertrain Engine Management System) for transmissions, ICE, HEV and BEV.

It is worth noticing that these elements belong to different nature fields (i.e. electronic, mechanic and so on). However, the thesis is focused on the team in charge of testing the electronic part production of these products.

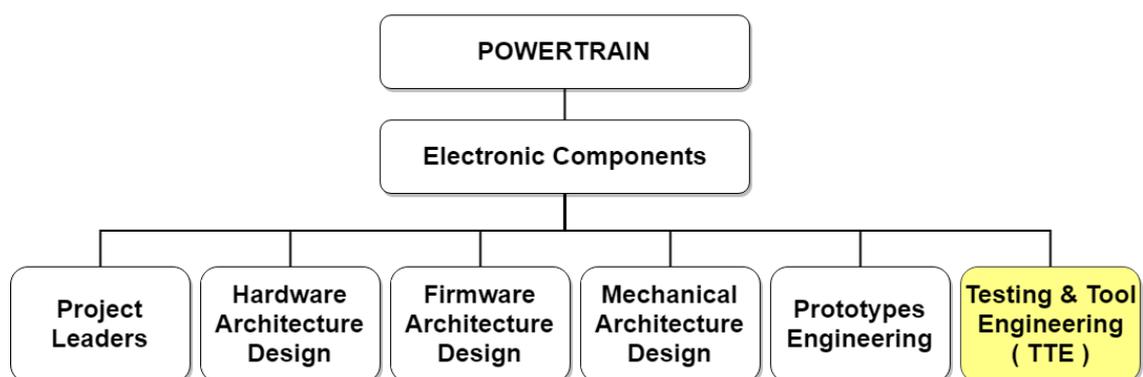


Figure 2: TTE - Position in the MM-PWT hierarchy

As [Figure 2](#) shows, the production of Electronic Components within the POWER-TRAIN business line involves different teams:

- **Project Leaders:** It is a cross-team composed by a few Professional Leaders endowed with a high level know-how to be spread with other employees.
- **Hardware Architecture Design:** It is the team in charge of performing the whole design process of the board, from specification to layout.
- **Firmware Architecture Design:** It is the team in charge of developing application and test firmwares in order for the board to run the SW application (during in-field operational life) and the test program (during testing phases).
- **Mechanical Architecture Design:** It is the team in charge of handling the mechanics of the electronic board such that connectors, seal to isolate the board, resin to isolate components and mechanical parts to be assembled to the board such as for example the throttle body¹ - gray color - shown in [Figure 3](#) (this mechanic body is assembled to the electronic board which in turn is isolated by the black seal. The board connector is visible on the bottom-left side).



Figure 3: Example - Electronically Controlled Throttle body

- **Prototypes Engineering:** It is the team in charge of retrieving prototypes. It is important to notice that prototypes are not produced by MM since the production centers work for huge amount of quantities. Thus, the production is entrusted to a supplier and an internal organization is needed to order, ship and retrieve the samples.

¹ The throttle body is assembled with a computer-controlled servo motor to move the butterfly valve according to inputs coming from the accelerator pedal. The movement of the valve regulates the amount of air that flows into the engine such that the air-fuel mix creates a gas useful for combustion.

[2]

The teams previously described, cooperate with the **Testing&Tool Engineering** team (yellow highlighted in [Figure 2](#)), which is in charge of testing the electronic part of all MM Powertrain business line products. The goal of the TTE-PWT team is to "provide to product development chain stable and reliable methodology and solution to check, validate and test final product, from the first prototype to mass production".

The team is structured as follows:

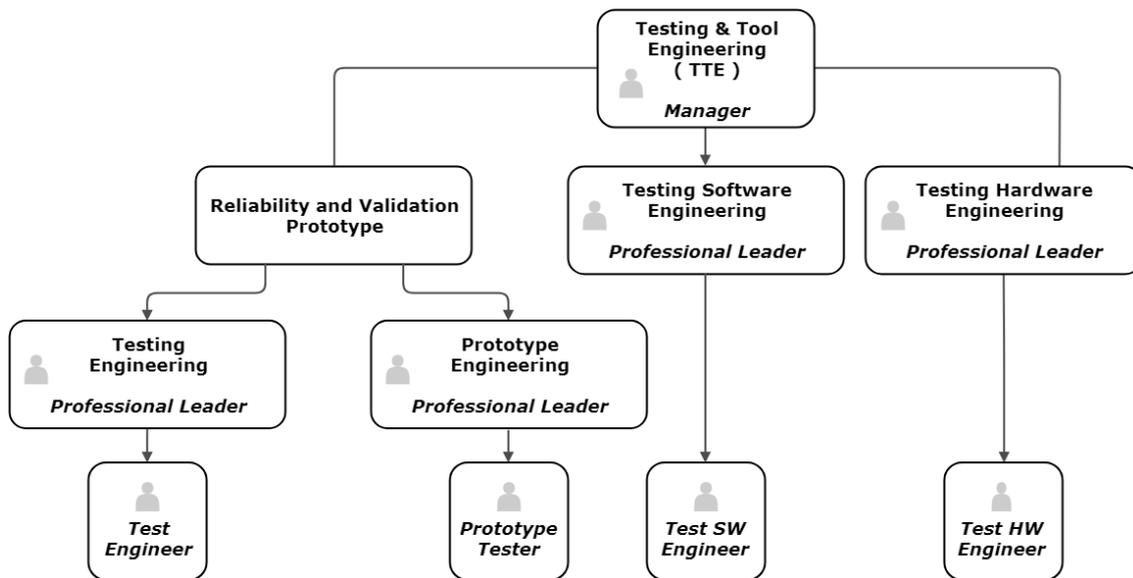


Figure 4: TTE - Team structure

Four sub-teams cooperate to fulfill the TTE-PWT mission. Each team is characterized by its own missions and responsibilities² :

- **Testing Engineering:** it involves the figure in charge of defining methodologies for the functional End of Line (EOL) test, developing EOL Test Program (TPGM) both for prototyping and industrialization phases, releasing coverage analysis specification and finally managing and supplying the test equipment on submitting periodical maintenance, calibration and fixture/harness design.
- **Prototyping Engineering:** it involves the figures in charge of applying functional TPGM sequences (developed by the Test Engineers) to a certain number of prototype samples such that verifying the reliability of the product before starting the industrialization phase (mass production).

² The team missions and responsibilities description refers to the production line adopted by MM-PWT. It is strongly suggested to consult [chapter 4](#) to fully understand these concepts.

- **Testing SW Engineering:** it involves the figure in charge of first verifying that, after production, the product passes ICT and EOL tests. Secondly, it prepares the board for in-field operational life by replacing the test firmware with the application firmware and writing into EEPROM the needed calibrations. The Test SW Engineer is also responsible, when a fail product is sent back to the company, for preparing it to be functionally tested by reloading the test firmware into it (since, in-field, it hosts the application firmware).
- **Testing HW Engineering:** it involves the figure in charge of applying stress tests to the products such as PRI³ (Power Run-In) in which some areas of the board are overstressed to verify how certain components react. These techniques are applied to both prototyping and returned-from-field samples.

The thesis, focusing on coverage aspects, has been developed in collaboration with the Testing Engineering sub-team. In the [section 2.3](#), the Test Engineer interactions are described in order to understand which relationships are needed to deal with coverage issues.

2.3 Test Engineer Interactions

The Test Engineer, while performing his tasks, needs to interact with the other Powertrain teams (reported in [Figure 2](#)) in order to retrieve and share information. These relationships involve:

- **TTE Manager:** to be aligned with shared planned activities.
- **Testing Engineering Professional Leader:** to be aligned with shared planned activities and receive technical advices related to choices to be taken during the workflow.
- **Hardware Designer:** to receive documents (such as schematic and layouts) needed to perform studies and analysis on the board before starting the test phases. Understanding how a product works is crucial for a Test Engineer in order to exercise proper functional tests on it.
- **Firmware Designer:** to receive the test firmware needed to be flushed on the board before applying the functional test.

³ It is the process through which components of a system are exercised prior to being placed in service. This testing process will force certain failures to occur under supervised conditions so that an understanding of load capacity of the product can be established. [27]

- **Mechanical Designer:** to receive specifics related to the connectors of the board such that retrieving in time cables and sockets to connect the DUT (Device Under Test) to the Automatic Test Equipment (ATE)⁴. Timing is crucial because, when a new product sample arrives, the ideal situation is having all material needed for the test to be performed.
- **Prototype Engineer:** to receive the prototype samples which have to be tested.

In the [PART II](#) the potential defects concerning the Powertrain electronic products and the test strategy line adopted by MM-PWT to detect them are described.

⁴ The functional test is performed by an ATE running a test program. The sequences of this program are based on the query/answer mechanism which requires that the board is connected with the ATE in order for them to communicate.

Part II

PRODUCTION TEST

PRODUCTION LINE

The [section 2.1](#) introduced the product manufactured by MM-PWT. This chapter reports a description of the MM-PWT production line ([section 3.1](#)) and an overview of the potential defects that can affect the final product ([section 3.2](#)).

3.1 From PCB to Final Product

The MM-PWT manufacturing line applies a transformation process to the PCB in order to obtain the final product (ECU/TCU). This section reports an overview of the whole production line in two macro phases: Soldering Process (Front End at [subsection 3.1.1](#)) and Mechanical Packaging (Back End at [subsection 3.1.2](#)). Within this process, several tests are performed (V-shape approach is adopted in order to allow a easy recovery to errors). However, among all these tests, only three test strategies have an impact on the hardware fault coverage (AOI, ICT and EOL). They are depth in [chapter 4](#).

3.1.1 FRONT END

In this phase, MM-PWT produces the PCBA which is nothing but a *"PCB that mechanically supports and electrically connects electronic/electrical components and connectors through copper lines, called traces. They run signals allowing the circuit board to function in a specifically designed way"* [32]. Practically speaking, in this phase the components are usually soldered to the PCB (thus obtaining the so called PCB-Assembled).

It is important to notice that MM-PWT designs the PCB whose production is entrusted to a supplier company. This means that the input product of this process is the PCB which, after soldering, will become the PCBA.

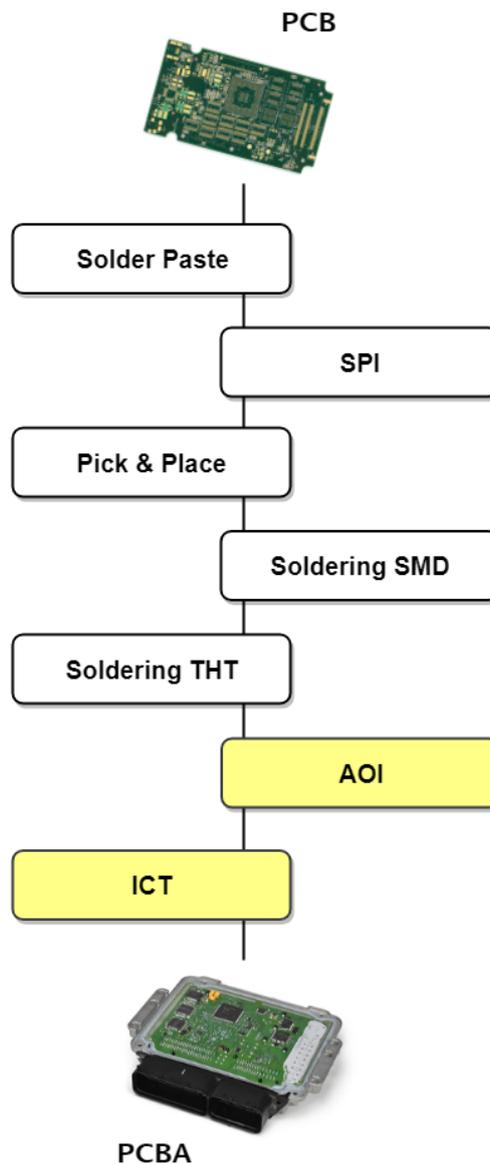


Figure 5: MM-PWT Production Line - Front-End

As the [Figure 5](#) shows, the MM-PCBA production line involves a sequential process [3] composed by the following steps:

1. **Solder Paste Stenciling:** The first step of the PCB assembly is to apply a solder paste in those places where components would be assembled to the board. The solder paste itself is a grayish substance consisting of tiny balls of metal (also known as solder). It is important to notice that the solder paste must be applied to the board at exactly the right places and in precisely the right amounts, otherwise unwanted situations may arise. An example of solder paste deposition is reported in [Figure 6](#), where the material is placed onto the pads that will host component leads.

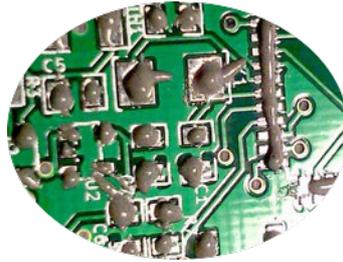


Figure 6: Production Process - Solder Paste Stenciling

2. **Solder Paste Inspection (SPI):** After the solder paste has been spread onto the board, before proceeding to the components placement, an optical inspection is performed to it such that verifying that solder paste has been deposited in right places and right amounts. The inspection consists in a automatic equipment which makes use of the image processing to detect unwanted situations. It is important to notice that a high number of faults could be avoided in this step before moving forward to the Pick and Place phase. This allows to fix the errors with very cheap solutions.
3. **Pick and Place:** After having applied the solder paste to the PCB and verified that no errors in the deposition occur, the process continues to the Pick & Place machine (Figure 7) which is a robotic device that places SMD components¹ on a solder-paste-prepared PCB (the placement coordinates are written in a BOM file - FUJI format²). In this step devices just lean to the solder paste. The effective solder consolidates in the next step (Reflow Soldering).

1 Components that are assembled to the PCB according to the SMT (Surface Mount Technology). This technology provides that components are soldered on the surface of the PCB [34]. Nowadays, PCBA hosts only SMD components except for connectors.

2 It is a text file reporting a list of components that should be assembled to the PCB. For each component, some information are reported. Among them there are also a couple of surface coordinates representing the position in which the component should be placed.



Figure 7: Production Process - Pick and Place Machine

As it can be noticed from the picture, the component reels are loaded into the machine which, after having been instructed, takes and places a precise component on a precise area.

4. **Reflow Soldering:** Once the solder paste and SMD components are all in place, they need to be stabilized. It means that solder paste needs to solidify, so that components adhere to the board. This is possible through a "reflow" process which occurs into a reflow oven which consists of a series of heaters capable of gradually heating the board up to temperatures around 250°, hot enough to melt the solder contained into the solder paste. Once the solder melts, the PCB continues to move through the oven. It passes through a series of cooler heaters which allows the melted solder to cool, thus solidifying in a controlled manner. This creates a permanent solder joints that connect SMD components to the PCB. It is important to notice that in the case of two-sided PCBA (board containing SMD components mounted on both sides), a double reflow steps arises, each for side separately, usually starting from the lighter side. The [Figure 8](#) shows an industrial reflow oven.



Figure 8: Production Process - Reflow Oven

5. **Through-Hole Component Insertion:** Previous steps, from 1 to 4 refers to the assembly of SMD components to the board. However, it can be possible that a board may involve another component technology different than SMT. These are the THT components³. In MM-PWT, board connectors belong to the THT category. The component leads, as reported in [Figure 9](#), are placed within holes that are present in the PCB to pass a signal from one side of the board to the other (VIAS).

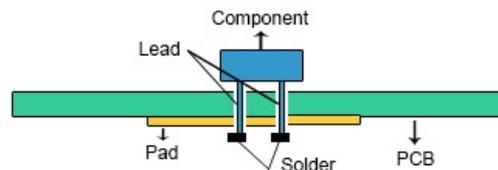


Figure 9: Production Process - THT Component

The THT soldering can involve different techniques:

- **Reflow:** It is similar to the process adopted for SMT. The solder paste is deposited on the PCB hole (which hosts the device lead) and the component is introduced into the oven. It is important to notice that the device lead should be properly placed in order to avoid loss of solder paste during reflow.
- **Local Wave Soldering:** It consists in a solder fountain on which the lead of the THT device is placed in order to be soldered. This technique is characterized by long performance times due to the fact that THT leads are soldered in series one after the other.
- **Press Fit:** This technique does not involve any solder material since the component is mechanically embed into the board. In fact, the device leads fit in the holes without needs of soldering process.

At the end of the THT soldering process, a solder like the one reported in [Figure 10](#) should be obtained. The yellow lead, crossing the PCB from one side to the other, is electrically connected to the board thanks to the grey solder substance consolidated by the process.

³ Components that are assembled to the PCB according to the THT (Through-Hole Technology). This technology provide that components are soldered through the holes of the PCB. Nowadays, PCBA hosts connectors as components belonging to the THT category. [35]

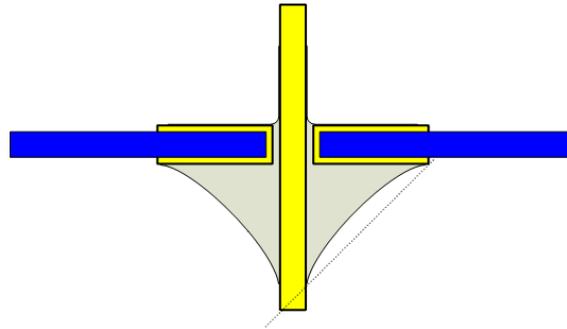


Figure 10: Production Process – THT Solder Geometry [1]

6. **Automatic Optical Inspection (AOI):** The soldering process, described in the previous steps, needs to be tested because it is important that *bad* boards do not go forward in the production process (proper fixing or discarding actions avoid this). The first test is related to a visual inspection in which a camera scans the DUT to detect catastrophic defects such as missing components, missing solders, poor quality of solders or components wrongly mounted. This technique is depth in [section 4.1](#).
7. **In-Circuit Test (ICT):** Once AOI passes, an additional test is performed, this time looking for electrical defects. It is performed by a special test equipment contacting the access points of the board, checking for short and open connections, passive and active faulty elements. This technique is depth at [section 4.2](#).

The process described above is performed for every PCBA to be produced. At this point, after all these phases are performed and passed, the product should be electrically unbroken.

From the [Figure 5](#), two test strategies are highlighted: AOI and ICT. These are the Front-End test strategies which impact on the hardware fault coverage. Both of them are depth in the [chapter 4](#).

3.1.2 BACK END

In the Front-End phase ([subsection 3.1.1](#)), the PCBA is produced and electrically tested to ensure that it has been assembled correctly. If these steps pass, the product is apparently electrically complete but actually it is not ready yet to operate in-field since mechanical protection against external influences are needed to guarantee that the board works without being disturbed by the surrounding environment. This is the reason why in the Back-End phase it is sealed with mechanical covers and finally functionally tested before being sent to the field.

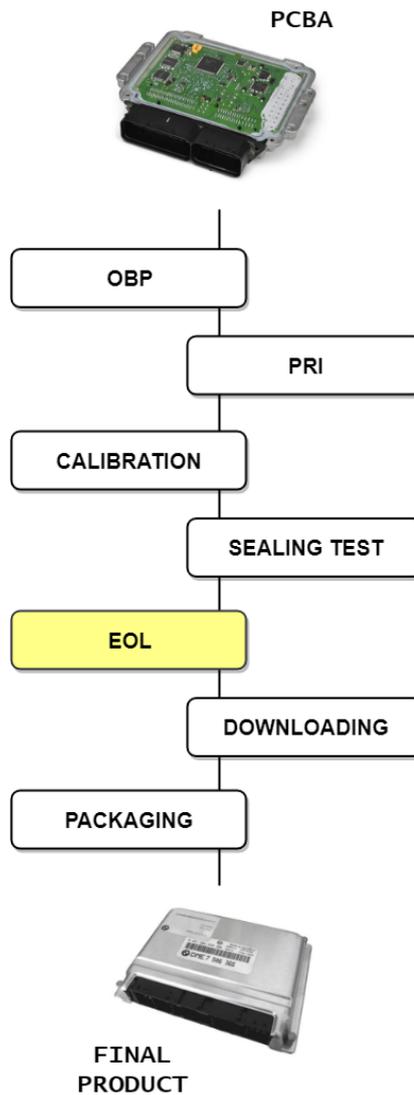


Figure 11: MM-PWT Production Line - Back-End

As the [Figure 11](#) shows, the MM-PWT Back-End involves a sequential process composed by the following steps:

1. **On-Board Programming (OBP):** During the Front-End phase ([subsection 3.1.1](#)) the MCU is virgin. In this phase it is flushed with the test firmware⁴ in order to prepare the board for the functional tests.

⁴ The test firmware makes available a routines interface layer based on a query/answer mechanism such that the test sequence invokes a routine and the MCU answers properly, in relation to the performed request.

2. **Mechanical Packaging:** Once the MCU has been configured with the test firmware, the product is ready to be subjected to some tests to understand whether it is ready to operate in-field or not. For this purpose, in this phase, the PCBA is packed into a mechanical cover which protects the board from external influences.
3. **Power Run-In (PRI):** It is a stress test, based on the fact that the probability for a failure to arise is characterized by a particular behavior, as reported in Figure 12. The curve resumes the fact that it is very likely that a product fails during the first phases of its life.

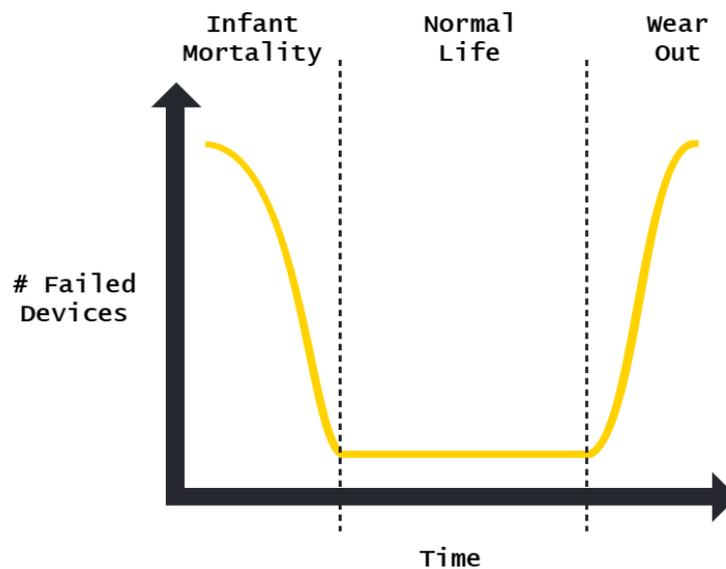


Figure 12: Infant Mortality - Bathtub Curve [37]

Thus, the purpose of PRI is to stress the product such that the first phases of its life are left behind and the product can proceed with future steps if during PRI no failures arisen.

4. **Calibration:** *"It is the procedure of reading offset and gain errors from the board and updating special onboard analog calibration circuitry that will correct these errors. When the board is calibrated, the calibration constants used to update the analog calibration circuitry, are stored in nonvolatile memory on the board. From memory they can be loaded when needed. The offset and gain errors, however, may change with time and temperature. As a result, the calibration constants, determined at the factory, may not apply to the current operating environment of the board. To achieve the specified accuracy of the board, it must be calibrated in the operating environment" [8].* MM-PWT computes these calibration at normal operating condition (such as ambient temperature 25°).

5. **Sealing Test:** This phase involves those tests which confirm that the board has been properly covered and isolated, ensuring the fact that it is not influenced by external entities. An example of sealing test provides that the product is dropped from a certain height in order to check whether it still works after the drop.
6. **End Of Line (EOL - Functional Test):** EOL is a test performed to the board at the end of the production line to ensure that, in addition to other passed checks, the product is also capable of performing its function. In particular, an automatic test equipment emulates the surrounding environment in which the board would operate in order to supervise the reactions to precise stimulus. This technique is depth at [section 4.3](#).
7. **Customer Software Download:** At this point of the production line, all tests performed on the product has produced a "PASS". It means that the product has been tested as much as possible and it is ready to work in-field. However, before being sent to the customer, the software application should be downloaded on the MCU to allow that everything works in a real environment. This is the Customer Software Download, through which the application firmware takes the place of the previous test firmware (needed to test the device).
8. **Packaging and Labelling:** In this phase, the product is finally packed and labeled. It will be successively sent to the customer.

In the [Figure 11](#) the EOL test strategy is highlighted since impacts on the hardware fault coverage. It is depth in the [chapter 4](#).

3.1.3 PROTOTYPES AND VALIDATION PHASES

The production line composed by Front-End and Back-End phases, described in [subsection 3.1.1](#) and [subsection 3.1.2](#), represents the standard flow which characterizes the production of an ECU/TCU.

However, design and production phases must have reached a certain stability before starting the mass production. For this purpose, the MM-PWT product evolves along the succession of three prototype versions ([Figure 13](#)) subjected to a waterfall validation process.

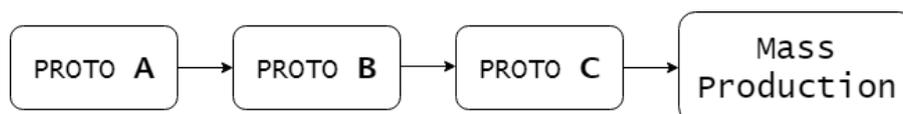


Figure 13: MM-PWT Production Line - Prototype Versions

The PROTO_A represents usually the previous release of the product (if it does not exist, a first draft of the product is projected and manufactured). The purpose of the PROTO_A is to provide a starting point to work on. After that some basic modification are applied to PROTO_A, the obtained PROTO_B is ready to be validated. At this point the design is questioned in terms of specification (DV): "*Is the design of the product compliant with the specifications?*". As soon as the answer to this question is positive, the activities of the R&D are frozen, the design and layout are declared to be stable and the final PROTO_C is produced. The production process of the PROTO_C is finally tuned (PV): "*Is the manufactured product ready to work as expected on the specified environment?*". As soon as the answer to this question is positive, the production is declared to be stable and the industrialization phase can start.

Note that prototypes are real boards and so their manufacturing follows all the steps involved in the production line. However, the Front-End, which is strongly dependent on the design and layout of the board, is set up only once that the PROTO_C is available (the product has reached a stable design version after DV). In fact, before that point, Front-End production is entrusted to a supplier company or otherwise some programmable technologies are used (as for example flying probe to perform ICT) such that being able to adapt to potential design changes.

3.2 Potential Defects

The PCBA production adopted by MM-PWT strongly relies on:

- **Supplier Company Production:** As anticipated, the PCB production (input of the process) is entrusted to a supplier company as well as some components to be mounted on board.
- **Proprietary Machinery/Personnel:** The transformation process applied to the PCB is a sequence of steps in which proprietary machinery and personnel involved have an impact on the consistency of the final product.

Thus, reliability of the final product is strongly dependent on a consistent number of factors. However, this complex process is not perfect (as everything in the world) and the final product can be affected by some defects. Next subsections describe two main categories of defects: those related to solders ([subsection 3.2.1](#)) and those related to components ([subsection 3.2.2](#)).

3.2.1 SOLDERING DEFECTS

Focusing on the soldering process (described at [subsection 3.1.1](#)), it is important to point out a concept: *"just because there is machinery involved does not mean that this process is less error prone than doing it by hand. In fact, soldering at the manufacturing level is a precise science that needs to be supervised"* [17]. The problems, otherwise, can be several:

- **Solder Bridges/Shorts:** It is an undesired short circuit between two points of the board, usually due to two solder joints that connect one to each other, as the [Figure 14](#) shows on the left side. This defect causes that the two pins are not independent anymore and failures can arise while driving or reading signals they carry.

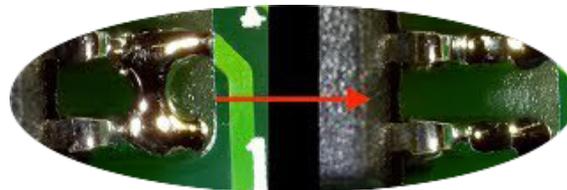


Figure 14: PCBA Defect - Solder Bridging

- **Solder Excess:** It is a excess of deposited solder during a THT reflow. The solder, which should be absorbed by the hole, remains on the top. Unfortunately there is no way to understand what is going on inside the ball. It can be possible, in fact, that the connection is not properly made (creating an undesired open circuit).



Figure 15: PCBA Defect - Solder Excess

- **Insufficient Solder:** It is a partial absence of solder in a connection point. The access to the points partially soldered is not always guaranteed. For example, in the [Figure 16](#), the VIA⁵ boundary is partially covered with solder (the red circle highlights a solder absence). If, for any reason, the ICT nail accesses

⁵ It is an electrical connection between layers in a physical electronic circuit that goes through the plane of one or more adjacent layers [36].

the VIA by contacting only the red-circled part, the electrical communication could not happen.

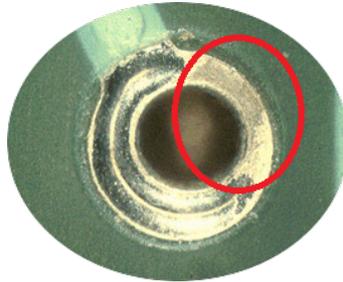


Figure 16: PCBA Defect - Solder Poor

- **Unwanted Solder:** It is an unwanted deposition of solder on the surface of the PCB after the wave soldering process. In fact it is expected that, during this time, only THT components are affected by the wave soldering method. However, due to some problems, certain areas are wrongly affected by solder deposition. This phenomenon is also known as "Solder balling". These unwanted solders are the highest causes of shorts on the PCBA.

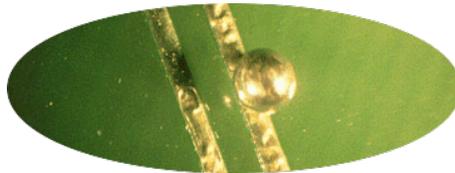


Figure 17: PCBA Defect - Solder Balling

- **Void Solder (Open Circuit):** It is a missed soldering point. This is a very critical problem since constitutes an unexpected non-connection point between PCB and component. As the [Figure 18](#) shows, the left pin of the component is not soldered to the PCB (no solder paste has been deposited before re-flow). However, an open circuit can be also generated by a bad soldering (as mentioned in "Solder Excess" defect).

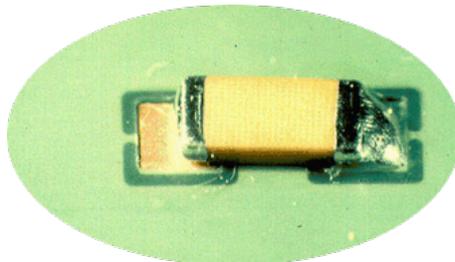


Figure 18: PCBA Defect - Solder Void

3.2.2 COMPONENT DEFECTS

Imprecision in the soldering process (described at [subsection 3.1.1](#)) or in the supplier silicon vendors devices production cause components to be affected by several defects:

- **Shifted Components:** It is a wrong alignment of the component concerning either the vertical or the horizontal axis.
A vertical-unaligned component can have a very low MTTF⁶ due to the fact that soldering are very weak and can easily deteriorate after being stressed a little. These considerations are related to THT components. [Figure 19](#) shows an example of vertical misalignment.

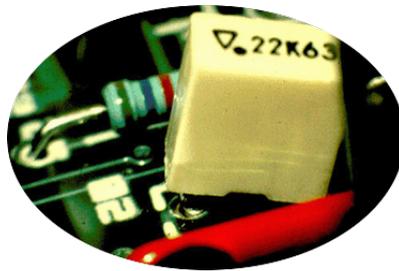


Figure 19: PCBA Defect - Lifted Component

However, the misalignment can be also horizontal as shown on [Figure 20](#).

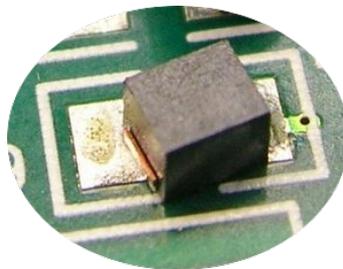


Figure 20: PCBA Defect - Shifted Component

- **Wrong Component:** All the above mentioned defects are strictly related to the soldering process. However, some other issues can be introduced by the Pick & Place step, as for example the "wrong component" defect. It is the case in which the Pick & Place places the wrong component. It is possible recognizing this defect only by looking into the reference number written on the top of the component.

⁶ Mean Time To Failure: it indicates the period of time before the component stop acting as intended.

- **Wrong Polarization:** The component is mounted reversely of a multiple of 90° due to Pick & Place issues. If the device is symmetric, there is no way to understand if its polarization is right unless looking into the polarization mark on the top of it, as yellow-circled in [Figure 21](#).



Figure 21: IC Device - Polarization Mark

It is important to point out that "Wrong polarization" refers to the fact that component is mounted in a wrong way. "Shifted", instead, means that the component is mounted in the right way but slightly shifted with respect to the position it has to occupy.

- **Missed Component:** It involves a situation in which the component, after soldering process, is not present at all. The issue is usually related to the Pick & Place machine which may miss the placement of the device on the board.
- **Faulty Component:** Due to errors arising during the device manufacturing, the component can be produced with internal defects and then sold to the customer. These defects may evolve in an undesired faulty behavior during the functional life of the product.

3.2.3 DEFECTS SUMMARY

Before proceeding to analyze the various test strategies within the MM-PWT hardware test coverage line, a general picture of the most important potential defects is summarized in this section such that later identifying, in [chapter 4](#), which defects could be covered by any technique.

As reported in [subsection 3.2.1](#), the most likely soldering defects are summed up in [Table 1](#).

SOLDER			
Short	Open	Excess	Insufficient

Table 1: Potential Defects - Solders Summary

As reported in [subsection 3.2.2](#), the most likely component defects are summed up in [Table 2](#).

COMPONENT				
Missed	Wrong	Faulty	Reversed	Shifted

Table 2: Potential Defects - Components Summary

A particular attention is reserved for BGA⁷ components whose assembly to the PCB still belongs to the SMT technology but, due to some solder ball spread on the bottom side, it is not easy to understand whether the reflow process ended well or not. In [Figure 22](#) it is possible to see that solder balls are hidden from the device.

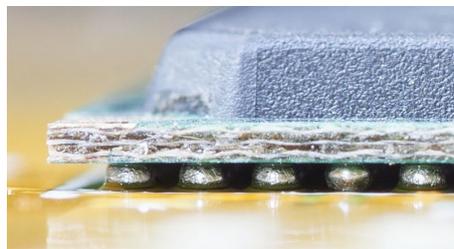


Figure 22: PCBA Defects - BGA Soldering

Thus, the most likely BGA defects are summed up in [Table 3](#).

BGA	
Short	Open

Table 3: Potential Defects - BGA Summary

⁷ "A ball grid array (BGA) is a type of SMD used for IC. BGA packages are used to permanently mount devices such as microprocessors. A BGA can provide more interconnection pins than can be put on a dual in-line or flat package. The whole bottom surface of the device can be used, instead of just the perimeter. The leads are also on average shorter than with a perimeter-only type, leading to better performance at high speeds." [26]

TEST STRATEGY LINE

The [section 3.1](#) describes the steps through which the bare board, received by the PCB supplier company, is transformed into MM-PWT final product passing through Front-End and Back-End phases. Due to imperfections of the process, some potential defects can be introduced to the product, as reported in the [section 3.2](#).

Responsibility of a test line is to achieve the highest capability in defects-detection, also called Hardware Fault Coverage.

This chapter focuses on those test strategies impacting on the final coverage: AOI ([section 4.1](#)), ICT ([section 4.2](#)) and EOL ([section 4.3](#)).



Figure 23: MM-PWT - Test Strategy Line

4.1 AOI

"Automated optical inspection (AOI) is an automated visual inspection of PCBA manufacturing in which a camera autonomously scans the DUT checking for both catastrophic failures (e.g. missing component) and quality defects (e.g. lifted components). It is commonly used in the manufacturing process because it is a non-contact test method." [25]

MM-PWT performs AOI in the post-reflow phase (as described in [subsection 3.1.1](#)) when solders are just consolidated in order to keep error-recovery costs as low as possible. In fact, potential defects could be early detected and quickly rectified.

In other words, AOI tests:

- Correctness and integrity of solders.
- Coherency between BOM FUJI and mounted components.
- Consistency of the component placements.

It is worth noticing that AOI uses image processing to verify the correctness of the reflow-process resulting product, as described in [subsection 4.1.5](#).

4.1.1 AOI EQUIPMENT

AOI is a test process which allows the operator to quickly and accurately ensures that PCBA under test have been correctly built without assembly errors. The AOI equipment that accomplishes this task is reported in [Figure 24](#).



Figure 24: AOI - Equipment

A common AOI system is composed by the following elements:

- **Monitor:** It is the mean through which the operator communicates with the machine. From the monitor the operator starts the board surface image acquisition and later checks that it does not present any defect (the acquired image is presented in a digital form allowing the operator to easily recognize undesired situations).
- **Lights:** The lights are used to illuminate in multiple color different angles of the board such that highlighting various details of the board surface at the moment in which the camera captures the image. Angles of the available colors can be customized regarding on the characteristics of the PCBA surface as well as the intensity of the light [4]. Cooperation of camera and lights is reported in [Figure 25](#). The camera in the middle acquires the image thanks to the surrounding lights which illuminate the target area.
- **Camera:** The camera is in charge of acquiring the image of a target area and sending it to the processing unit which, in turn, makes available it to the operator in a digitized format. The better the cooperation of the lights with the camera, the more detailed the acquisition of the product surface. Camera surface acquisition process is reported in [Figure 25](#).

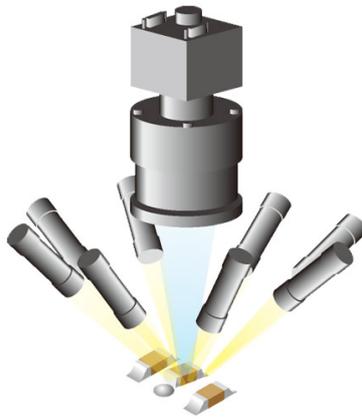


Figure 25: AOI - Camera and Lights cooperation [38]

- **Storage/Backup:** A storage/backup system is needed if the "*Statistical pattern matching*"¹ is performed. This techniques, adopted to inspect whether any defect is present, is based on previous history camera acquisitions.

¹ Statistical pattern matching technique provides that the acquired image is compared with all occurrences of a historical database containing images of several boards and several types of failure in order to identify defects on the product. [14]

- **Processing Unit:** It is the core of the system which receives the camera acquisitions and processes the digitized image of the entire board surface, possibly highlighting potential defects. This resulting image is shown up on the monitor in order for the operator to inspect whether the highlighted defects are actually present on the PCBA or not.

4.1.2 AOI TECHNIQUES

This section describes the "Acquisition" ([subsubsection 4.1.2.1](#)) and "Comparison" ([subsubsection 4.1.2.2](#)) techniques adopted by Automated Optical Inspection test strategy in order to detect and highlight any defect or suspect area on the soldered PCBA surface [14].

4.1.2.1 Acquisition

In this phase, the AOI equipment builds up a picture of the board which is preparatory for the next phase: Comparison ([subsubsection 4.1.2.2](#)).

Basically the board is submitted to several lights while the cameras are capturing the images of the surface, as described in [subsection 4.1.1](#). Through some image processing techniques the various captured images are merged to build up the picture of the whole board.

An important trade off speed/accuracy characterizes the image capture system. In fact, different techniques could be used such as "Streaming video" and "Still image camera system". In the former, the camera takes a streaming video from which captured frames are subjected to image processing in order to generate the final surface image. This technique is very fast but on the other side low accurate. In the latter technique, the camera is taken very close to the surface as well as the lights which play an important role. The resulting image is very accurate and rich of details even though the process to move the camera, controlled via software, is very time-consuming.

Another important role is played by the lighting system that, if accurately chosen, allows to highlight different kinds of defect more easily. The most used lighting systems are fluorescent, LED and infra-red/ultra-violet ones.

4.1.2.2 Comparison

After "Acquisition" phase ([subsubsection 4.1.2.1](#)), the captured image is compared with the knowledge of the machine concerning how the board surface should appear. As a result of this comparison, AOI is able to detect and highlight unwanted situations.

Different strategies are used to compare the actual board image acquisition to the "golden" one:

- **Template matching:** AOI system uses a "golden board" image present in its storage to make the comparison.
- **Pattern matching:** AOI system compares the actual board image to a static set of stored pattern figures related to both good and bad PCB assemblies.
- **Statistical pattern matching:** AOI system compares the actual board image to an averaged figure related to a set of previous acquisition of boards and failures. The advantage in this case is that the storage learns increasingly time after time, making the averaged figure very reliable.

The more accurate the adopted comparison technique, the higher the AOI capability of detecting lack or excess of solders as well as missing components or whatever concerning soldering process defects.

4.1.3 AOI LIMITS

Physical objects can be visually inspected in the sense that the way they look like can be compared to an ideal representation of how they should be. However, an image is only an observation which hides several aspects of a board such as behavior of the devices, hidden parts of the PCBA and functional behavior of the product. This is the reason why AOI is a technique looking for catastrophic situation, for which a repair of the board is needed before proceeding.

In addition, it is important to notice that AOI only assists the operator in recognizing defects. The downside of this aspect is that the capability of the inspector after a certain amount of work can be reduced drastically (the operation to be performed never changes and attention of the user decreases) [19].

4.1.4 AOI DETECTION CAPABILITIES

On the basis of the AOI testing techniques described in the [subsection 4.1.5](#), it is possible to digest the capabilities of Automated Optical Inspection strategy concerning the detection of the most likely defects summarized in [subsection 3.2.3](#).

<i>AOI</i>	
<i>SOLDER</i>	<i>Short</i>
	<i>Open</i>
	<i>Excess</i>
	<i>Insufficient</i>
<i>COMPONENT</i>	<i>Missed</i>
	<i>Wrong</i>
	<i>Faulty</i>
	<i>Reversed</i>
<i>BGA</i>	<i>Short</i>
	<i>Open</i>

Table 4: AOI - Defects Detection Capabilities

The [Table 4](#) broadly highlights what AOI is capable of detecting in terms of most likely defects.

Concerning **Solder** defects, they are all detected by AOI during Comparison phase as described in [subsection 4.1.2.2](#).

Concerning **Component** defects:

- **Missed:** Detected by AOI during Comparison phase as described in [subsection 4.1.2.2](#). This is a great feature concerning AOI since it is a substantial validation before proceeding to a second electrical check of the board (ICT - [section 4.2](#)).
- **Wrong:** Undetected by AOI since the particular MM-PWT inspection looks for soldering defects and does not compare any component label checking whether the BOM is consistent with the actual assembled board (this feature is called OCR - Optical Character Recognition).
- **Faulty:** Undetected by AOI since it is a soldering-process verification based on a visual inspection. Any functional test is performed on the board.
- **Reversed:** Detected by AOI if and only if the *Polarity Check* feature, which checks for any component mark on the component body, is implemented. In MM-PWT this is not always the case and so this defect can not be considered always covered.
- **Shifted:** Detected by AOI during Comparison phase as described in [subsection 4.1.2.2](#). It is important to point out that multiple angular acquisitions

must be performed as clarified in [Figure 26](#) in which a lifted component, not detected in a first top acquisition (left picture), is detected in the successive angular ones (middle and right pictures).

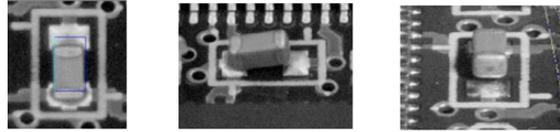


Figure 26: AOI - Lifted Component detection [7]

Concerning **BGA** defects:

- **Short**: Undetected by AOI since the camera scans the board surface not being able to look at the covered (not visible) places (as for example BGA balls bridged to each other).
- **Open**: Undetected by AOI since the camera scans the board surface not being able to look at the covered (not visible) places (as for example BGA ball absences).

4.1.5 AOI FUTURE

To conclude, the incoming increased productivity of the array components (e.g. BGA) is creating more and more areas of the board not visible from AOI equipment. This may bring the production line to migrate more and more towards AXI technique which is, in the same way, a visual inspection technique with the additional advantages of facing some AOI drawbacks (as for example the detection of defects concerning BGAs). This is allowed by X-ray acquisitions that capture the status of the solders even though these are hidden. An example of AXI acquisition of BGA solder balls is reported in [Figure 27](#).

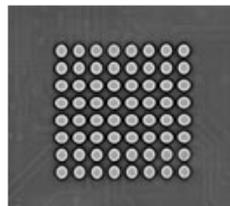


Figure 27: AXI - BGA defects detection [15]

It is important to notice that AXI is not a real future aspect but it can be considered an evolution of AOI since, in MM-PWT, a high number of production lines already have the AXI technique.

4.2 ICT

"In-Circuit Test (or also ICT) is an example of white box testing in which a PCBA (Printed Circuit Board Assembled) is tested by electrical probes checking for shorts, opens, resistances, capacitances and all other quantities that show whether the PCB was correctly fabricated and assembled" [28]. It may be performed with a bed of nails type test fixture cooperating with a specialist test equipment, or with a fixtureless in-circuit test setup.

MM-PWT performs ICT Bed of Nails in the post-reflow phase during Front-End process (as described in [subsection 3.1.1](#)) to complete the electrical tests of the PCBA before proceeding with the functional checks (in the Back-End process [subsection 3.1.2](#)).

In other words, ICT tests:

- PCB integrity and functionality.
- Passive components soldering and positioning.
- Passive components value and tolerance check.
- Active components mounting and basic functionality, when possible.

It is worth noticing that ICT does not give a test of the functionality of a board, but it checks whether it has been assembled correctly. In fact, the test equipment operates by measuring each component in order to check whether it is in place and it has the correct value.

4.2.1 ICT EQUIPMENT

In-Circuit Test is performed by a programmable test equipment. This section reports a description of the hardware architecture of the In-Circuit test system. An ICT system is schematized as reported in [Figure 28](#).



Figure 28: ICT - Equipment

A common ICT system is composed by the following elements:

- **PC:** The PC is used to handle the operation concerning the ICT test to be performed (run, check results, program the test sequences to be applied and so on so forth). It is often a general purpose computer, equipped with a commercial operating system, running the ATE software environment from which the user can control the ATE hardware.
- **ATE:** The Automated Test Equipment involves all equipment needed to perform the test of a board such as the analog/digital instruments, power supply blocks and channels to connect the ATE instruments to the ICT tester nails.
- **Bed Of Nails:** It is an electronic test fixture with a high number of nails that would contact the test points of the Device Under Test. In turn, each nail is multiplexed to an ATE instrument in order to stimulate and sense a signal to/from the DUT. The advantage of a Bed of Nails ICT is that all test points of a board can be contacted at once (the variant, Flying-Probe, is a dynamic fixture which is composed by few nails moving on the PCB surface to perform sequential tests on the board).

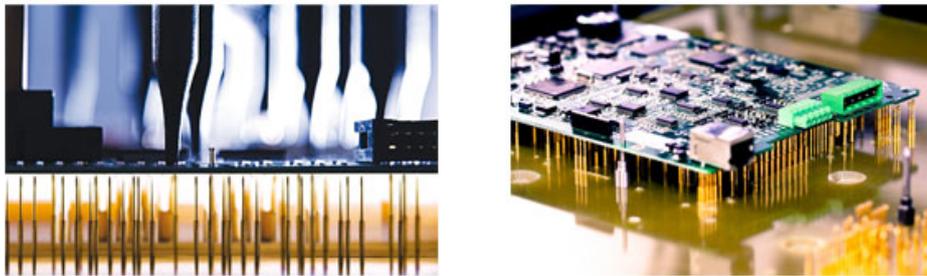


Figure 29: ICT - Bed Of Nails [16]

As the [Figure 29](#) shows, the fixture is composed by a flat brown support with a certain number of holes. These holes allow nails to remain orthogonal with respect to the board to be accessed (as shown on the left). In an idle position, as reported in the figure, nails do not contact the board. However, when pressed down, the DUT is this time accessed through access points / nails contact.

- **ATE / Bed Of Nails communication system:** It is important to notice that regarding on the kind of stimulus or sensing to be performed, nails should be multiplexed with precise instruments within the ATE. Connection instruments-nails change during the ICT test is being performing thanks to a program-controllable system of relays and switches that can be opened and closed by program instructions. The ATE-Bed Of Nails interconnection system is schematized in [Figure 30](#).

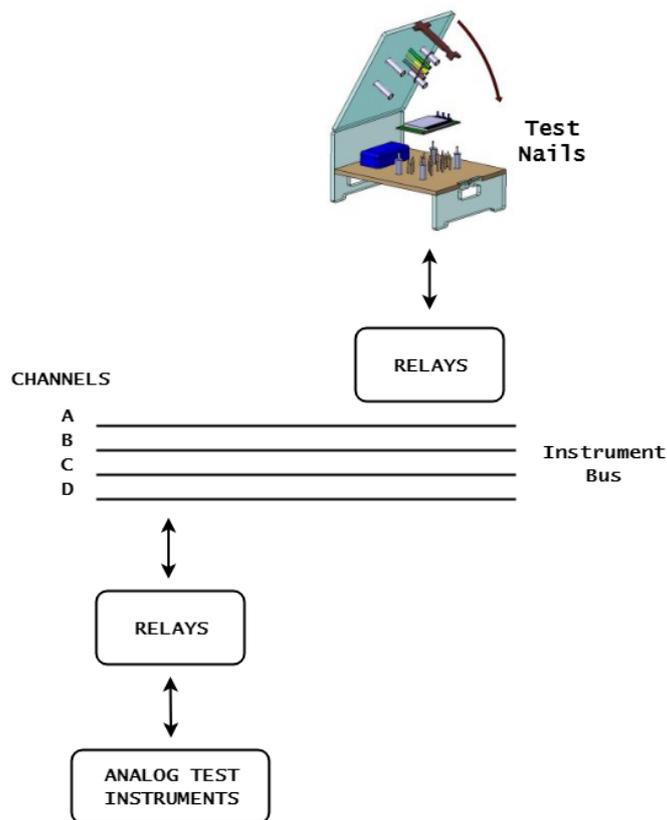


Figure 30: ICT - ATE-Bed Of Nails communication system

Analyzing [Figure 30](#), starting from the bottom side, a set of test instrument blocks are present (i.e. DC Current/Voltage Source, DC Voltmeter, DC Ammeter and AC Impedance Measurement Module). These instruments belong to the ATE. They are in a first phase multiplexed to the ATE channels through a programmable relays system. Once that test instruments are multiplexed to the channels, a final step called "scanner" is programmed such that picking up signals coming from channels and bringing them to the desired nails in order to stimulate or sense precise elements of the board.

It is worth noticing that in an ideal situation, involving an infinite-channels ATE, the first multiplexing phase can be skipped since there is a channel for any instrument.

- **DUT:** It is the just produced PCBA to be tested.

4.2.2 ICT TECHNIQUES

This section describes the whole ICT test flow, composed of several sub-tests performed in order [13]:

1. **Pre-Screening Test:** It involves those tests looking for short ([subsection 4.2.2.3](#)), open ([subsection 4.2.2.2](#)) and non-contacting nail ([subsection 4.2.2.1](#)) defects on the entire board.
2. **Passive Component Test:** The passive components test is performed on the unpowered board in order to detect the presence and measure the value of some components such as resistors ([subsection 4.2.2.4](#)), capacitors ([subsection 4.2.2.5](#)), inductors ([subsection 4.2.2.6](#)), diodes ([subsection 4.2.2.7](#)) and IC components presence ([subsection 4.2.2.8](#)).
3. **Discrete Component Test:** The discrete component test is performed on the unpowered board in order to detect the discrete behaviors of certain components. An example of discrete test is "Transistor test", reported in [subsection 4.2.2.9](#).
4. **Special Test:** Special tests are dedicated for those components that require particular stimulus and measurements scenario to be performed, different than the commonly used routines.
5. **IC Test (Supplied Test):** The supplied test provides that the board is powered on to verify correct functioning of active components. It is important to notice that "supplied tests" can belong to the category of "special tests" also and vice-versa.

The ICT standard flow uses the "growing power" technique which provides that first phases are characterized by low level of voltages which step by step grow up until reaching last phases (where usually V_{batt} is reached). The level of voltages in the flow are reported in [Figure 31](#). An exception is made for the special tests since the level of voltage depends on unit to be tested and adopted techniques.

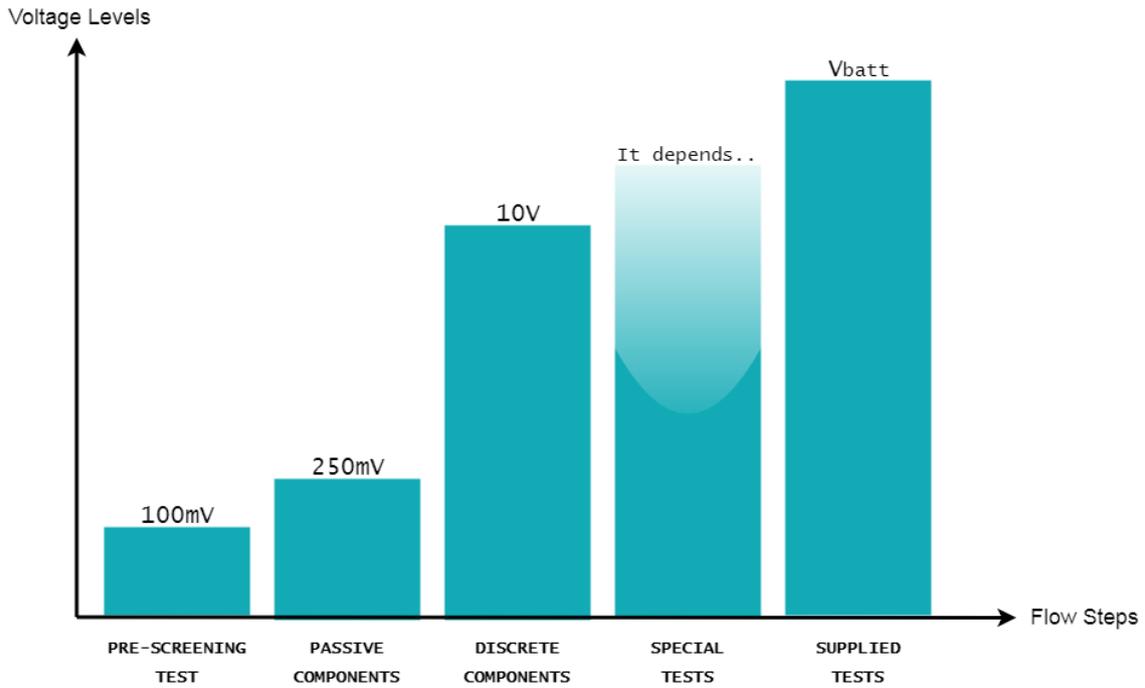


Figure 31: ICT - "Growing Power" flow

The ICT flow reported in Figure 32 is nothing but a sequence of routine calls. The executable program, running on the ATE, launches these tests such that the nodes of the board are properly stimulated and sensed. If all sub-tests pass, the board is considered to be "good". Otherwise, at the first fail, the board is considered to be "bad".

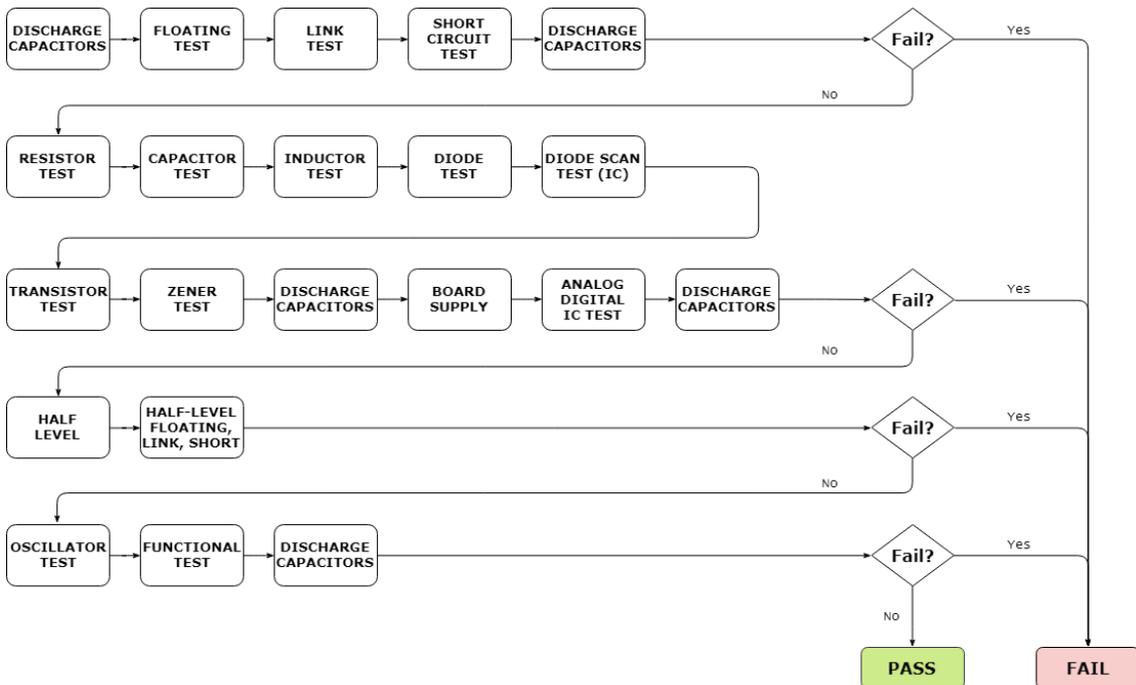


Figure 32: ICT - Test Sequences Flow

It can be noticed from [Figure 32](#) that two particular routines are several times invoked: "DISCHARGE CAPACITORS" and "STOP ON ERROR" (the ones with the label "FAIL?"). The former is present to avoid residual voltages at the end of those tests that provide injection of currents (the disadvantage is that a certain amount of time is required to perform this macro). The latter one, "Stop on error", is due to a safety-choice based on the fact that moving forward after a failed sub-test can compromise the integrity of the board.

Next subsections describe how some tests within the flow are performed.

4.2.2.1 Floating Test

It detects whether there is any nail which is not contacting the DUT. It is performed by measuring with an Ohmmeter the impedance between one TP and all others. The test is repeated for any TP. A high impedance threshold is taken into consideration such that if the measured impedance is higher than this threshold, the nail is considered to be not in contact with the TP.

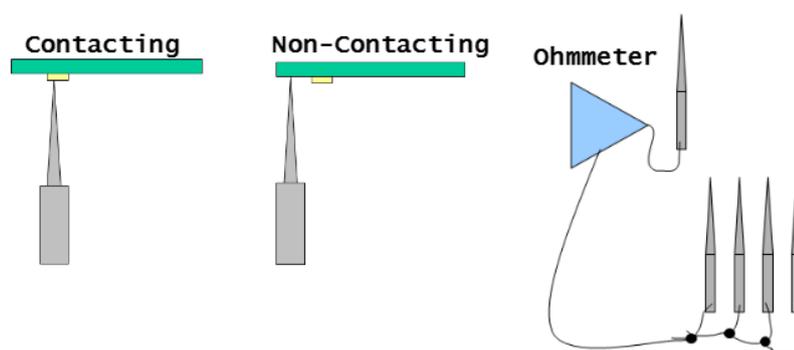


Figure 33: ICT Technique - Floating Test

From [Figure 33](#) it is possible to see two situations: one in which the nail is not contacting with the TP (in that case the impedance measured between that TP and others is very high) and the other in which the nail is contacting with the TP (in this case impedance depends on the link-relation of the TP with other TPs but it is not surely bigger than the adopted threshold).

4.2.2.2 Link Test

It detects if chained-TPs² are in contact between them. It is performed by injecting a current and measuring a voltage. If applying the Ohm's Law (voltage measured

² Chained Test Points are those connected together with a link-impedance very low. They are usually interconnected through jumpers or other low-impedance elements.

divided by current injected) the impedance between two chained TPs is higher than a threshold, some open circuits within the chain are detected.

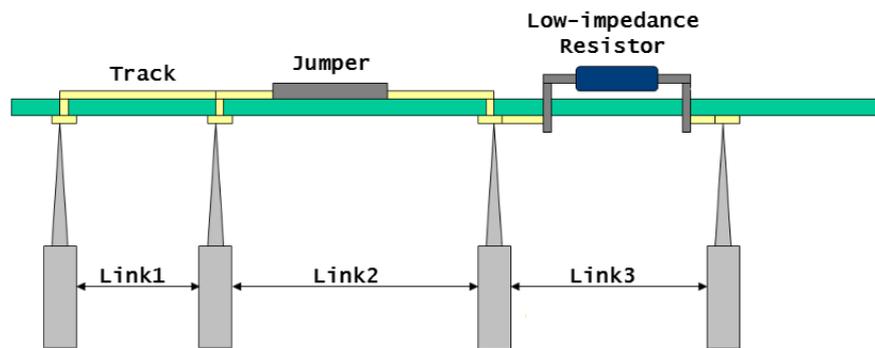


Figure 34: ICT Technique - Link Test

From [Figure 34](#) it is possible to see that three links are taken into consideration. The impedance measured at the ends of each link should be lower than a precise threshold, otherwise some missed components are detected.

4.2.2.3 Short Circuit Test

It detects unexpected low-impedance (short circuit) between TP/TP, TP/chain and chain/chain. It is performed by measuring the impedance with an Ohmmeter between one chain (head-chain) and all others. The test is repeated for all chains, changing any time the head-chain. If the measured impedance is lower than a threshold, a short circuit is detected.

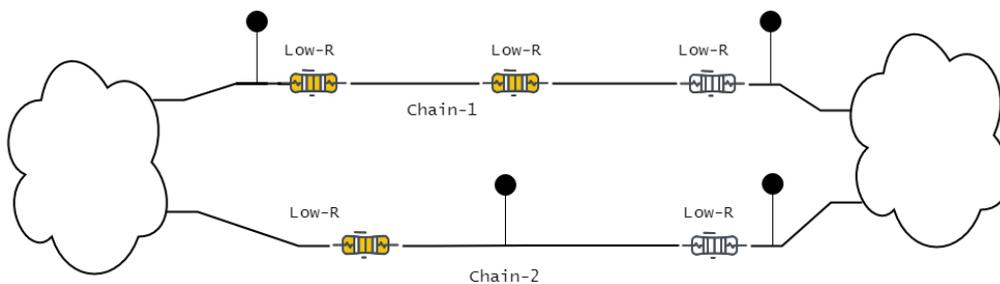


Figure 35: ICT Technique - Short Test

In [Figure 35](#) it is possible to see two chains, apparently independent in the sense that the impedance between them is not easy to be computed but it is surely not small. If for any reason, the computed impedance between these two chains is smaller than a precise threshold, a short between these chains is detected.

4.2.2.4 Resistor Test

It aims to measure the value and detect defects related to all resistors of the board. It is performed by injecting a current in order to measure a voltage. Through the Ohm's Law ([Equation 1](#)) the resistor value is then computed.

$$V \equiv R * I \quad (1)$$

It is crucial to understand that while testing a passive component some issues have to be considered:

- **Component Isolation:** The component to be measured must be isolated from the rest of the circuit since the injected current should flow only through the resistor under test in order for the measurement to be as precise as possible. If the component is not properly isolated, the injected current also flows in other branches, and the measured resistor value results smaller than the real one. The technique implemented by the ATE used to isolate components is called "Guarding". It is based on a voltage follower (low input impedance, high output impedance) that is used to bring to 0V the voltage drop between the ends of the branches to be isolated such that no current flows through them. An example to clarify the concept is reported in [Figure 36](#).

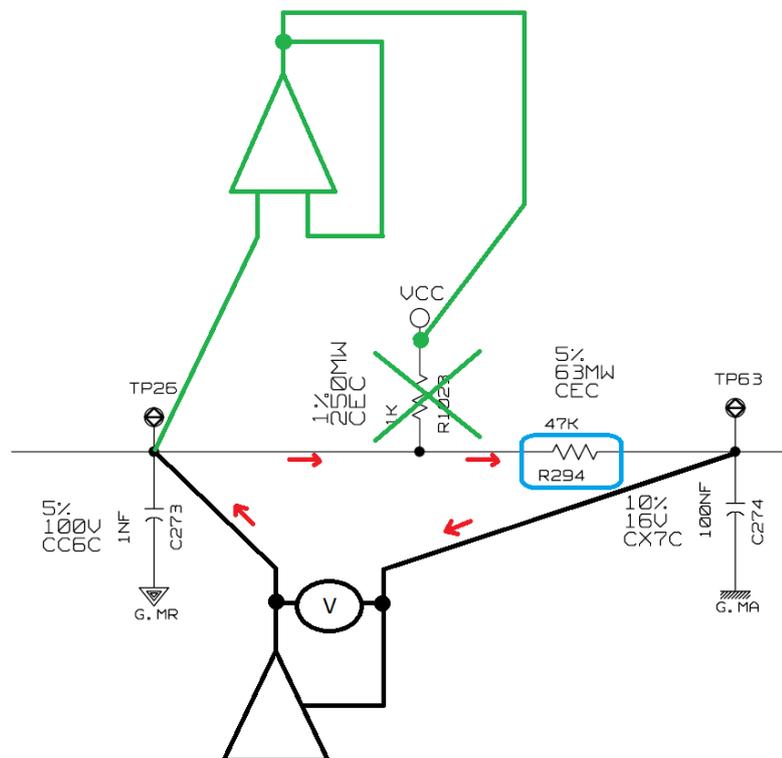
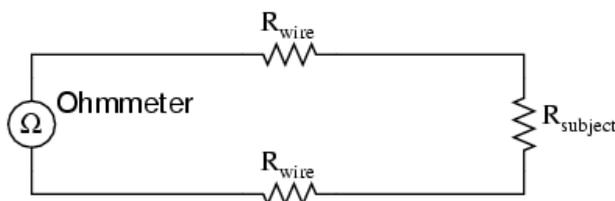


Figure 36: ICT Technique - Analog Guarding Technique

Analyzing the [Figure 36](#), the resistor R294 is the subject of the test (blue-circled). The "RESISTOR TEST" routine has the role of checking its presence and measuring its value. In order to do this, a current is injected with the ATE current driver (the black in the figure) in order to measure the voltage between TP26 and TP63 such that applying Ohm Law and computing resistor value. The problem is that the injected current can flow through another branch toward VCC. The guarding technique avoid the flowing current on this branch by using a voltage follower (the green in the figure) which brings the voltage on the node TP26 to the node VCC such that the voltage drop in the undesired branch is now 0V and no current flows in that branch anymore. Moreover, intrinsically, the voltage follower is characterized by having high output impedance level. This helps avoiding the current to flow into it. At this point, after "Guarding" is applied, the injected current flows exclusively through the R294 element allowing to rightly measure its value.

- **Low value Resistor** : When the value of the resistor is very low, some problems arise due to the fact that the connections between the ATE and the DUT introduces certain non-negligible levels of impedance (R_{wire}) that cause the resistor to be considered with a higher value with respect to its nominal one, as the [Figure 37](#) explains.



Ohmmeter indicates $R_{wire} + R_{subject} + R_{wire}$

Figure 37: ICT Technique - 2-wires Resistor Test

The Ohmmeter is composed of a current driver and a voltage measurement. This technique is called "2-wire" due to the fact that just two lines are used to connect the ATE to the DUT (current driver and voltage measurement are placed in parallel to compose the Ohmmeter). In this way, the current injected by the ATE current driver creates a voltage drop on both the DUT resistor to be measured and on the impedance created by the ATE-DUT connection lines. All these drops are taken into consideration by the voltage measurement. Thus, the computed resistance value would wrongly involve both R_{wire} and $R_{subject}$. It is possible to face this problem by using a 4-wire measurements, called also "Kelvin" technique, as the [Figure 38](#) reports.

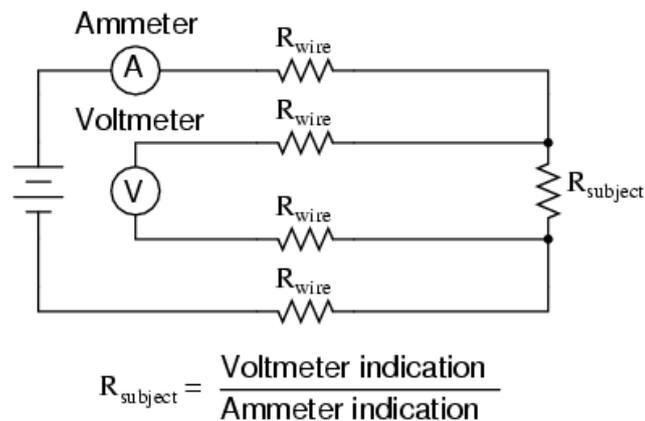


Figure 38: ICT Technique - 4-wires Resistor Test (Kelvin Technique)

It can be noticed that now 4-wires are used: 2-wire to connect the current driver and 2-wires to connect the voltage measurement to the resistor pins. The advantage here is that the voltage drop caused by the current injection is not taken into account by the measurement which, instead, rightly considers only the voltage drop on the R_{subject} . However, the disadvantage is that 4 TPs must be present on the board to allow such a measurement (2 TPs per pin).

4.2.2.5 Capacitor Test

It aims to measure the value and detect defects of all capacitors. It is performed by injecting a constant current in order to measure a voltage difference perceived during the time period in which the constant current is applied. Through the Coulomb's Law ([Equation 2](#)) the capacitor value is then computed (it is graphically the slope of the measured voltage), as the [Figure 39](#) reports.

$$I(t) \equiv C * \frac{dV(t)}{dt} \quad (2)$$

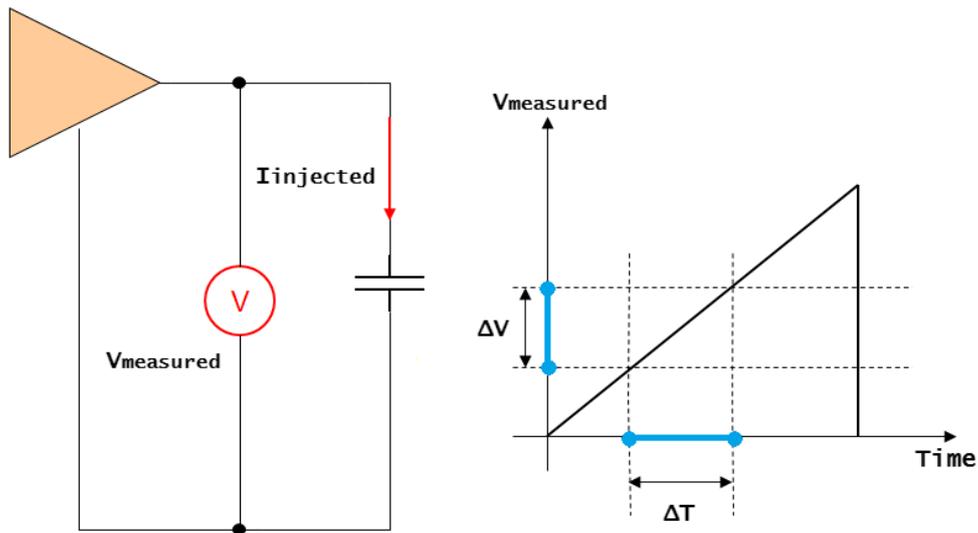


Figure 39: ICT Technique - Capacitor Test

The same issues, already described in the [subsubsection 4.2.2.4](#), are solved adopting the same techniques.

4.2.2.6 Inductor Test

It aims to measure the value and detect defects of all inductors. It is performed by applying a voltage difference at the inductor pins for a certain period in order to measure the current variation. Through the Lenz's law ([Equation 3](#)) the inductor value is then computed (it is graphically the slope of the current ramp), as the figure explains.

$$V(t) \equiv L * \frac{dI(t)}{dt} \quad (3)$$

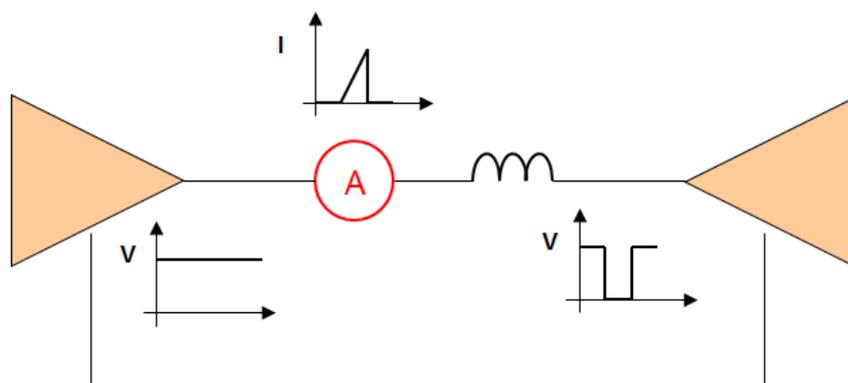


Figure 40: ICT Technique - Inductor Test

Analyzing the [Figure 40](#), the driver on the left is kept at a constant voltage. The driver on the right is made varying such that, at a certain point, a voltage drop at the inductor ends is created. This generates an increasing current whose slope is then measured.

The same issues, already described in the [subsubsection 4.2.2.4](#), are solved adopting the same techniques.

4.2.2.7 Diode Test

It aims to measure the value and detect defects of all diodes. It is performed by injecting a direct current in order to measure the V_{forward} , as reported in [Figure 41](#).

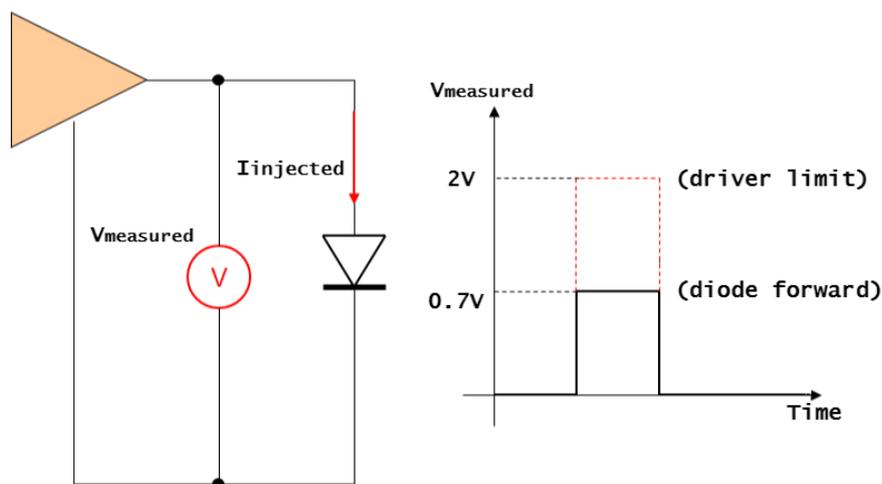


Figure 41: ICT Technique - Diode Test

The basic of this concept is to consider the diode as a voltage limiter. This is because a current driver, working at maximum 2V, generates a direct current to bring the diode in forward configuration. In this way, if the diode is present, it leaves on its ends a voltage drop of 0.7V (forward voltage of the diode). Otherwise, the voltage measured by the Voltmeter is equal to the maximum voltage that can be driven by the current driver (2V).

4.2.2.8 Diode Scan Test

It aims to detect defects related to IC pins (pins not soldered well). It is based on the idea that functional pins in an IC are usually connected to GND or VDD through the so called "clamping diodes" whose function is to prevent IC pins from voltage peaks. Thus, this test is performed by looking for a diode between the pin and GND or VDD. The concept is similar to "Diode Test" ([subsubsection 4.2.2.7](#)): a current is injected to bring the eventual diode in forward configuration such that measuring V_{forward} (0.7V).

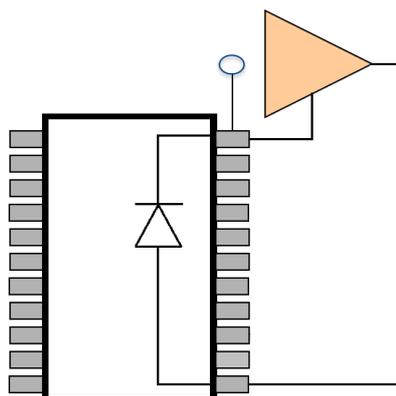


Figure 42: ICT Technique - Diode Scan Test

It is worth noticing that if the test is not able to read a forward voltage drop, it does not mean that the related pin of the IC is not soldered well. The test is used just for a confirmation that the pin of the IC has been soldered well since not all pins are protected by clamping diodes. However, even though a clamping diode is present, some limitations, as reported in [subsubsection 4.2.3.2](#), characterize this technique.

4.2.2.9 Transistor Test

It aims to check the MOSFETs behavior in both ON and OFF operation modes. It is based on the idea that while the MOSFET is OFF it works as an open circuit and when it is ON it is characterized by a V_{ds} drop on its pins. Thus, the transistor test is performed by stimulating the switching from OFF to ON and checking whether the measured voltage between drain and source (V_{ds}) changes level, as reported in the [Figure 43](#).

Analyzing [Figure 43](#), the driver on the left passes from low to high voltage level such that the transistor configuration passes from OFF to ON (driver on the right is kept at fixed voltage. The variation in the gate-source voltage drop should produce a transistor switching). If the transistor is not present or it is damaged, when the driver-left passes from low to high, the measured voltage remains stuck at the right-driver voltage limit.

It is also worth noticing that "Transistor test" belongs to the "discrete test" category since voltages applied are in the order of 10V approximately (as the [Figure 31](#) reports), high enough to bring transistors in saturation.

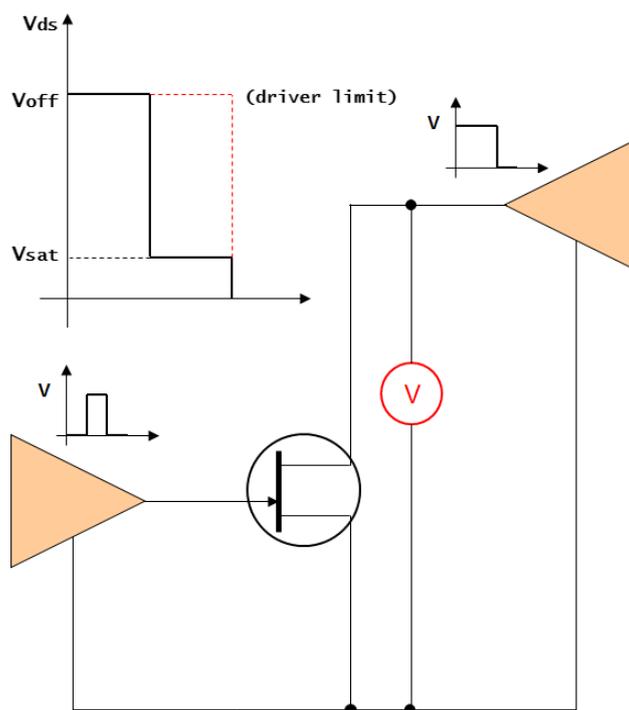


Figure 43: ICT Technique - Transistor Test

4.2.2.10 Operational Amplifier Test

It aims to test whether the OA rightly saturate during the low-high and high-low output transitions. The test is performed by powering up the OA and driving both its inverting and non-inverting pins such that the low-high and high-low output transitions are excited. The [Figure 44](#) describes how the test is performed.

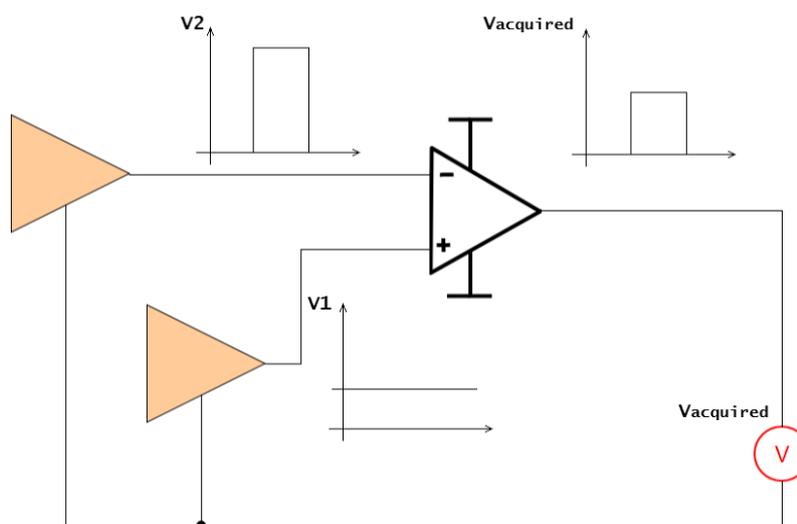


Figure 44: ICT Technique - Operational Amplifier Test

In the [Figure 44](#) it can be seen as both saturation outputs (high and low) and both transitions (high-low and low-high) are excited by creating an input voltage difference firstly negative and secondly positive ($V_2 - V_1$) generating a low and high saturation respectively.

This can be considered a test belonging to both categories of "discrete" and "supplied" tests since the technique used to test the component is specific for the operational amplifier class and also because the OA is power supplied in order to be tested.

4.2.2.11 Digital Test

Digital components are tested by exercising precising input combination in order to detect potential output misbehavior. In the [Figure 45](#), an example of digital test applied to an AND gate is reported. The gate is excited with all four possible input combinations (columns XY in [Table 5](#)) and the output is compared with the expected one (column Z in [Table 5](#)).

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

Table 5: ICT Technique - Digital Test patterns

However, as in the analog domain, it is important that the portion of digital circuit is isolated to avoid that surrounding elements interfere with the test. For this purpose the "guarding" technique is used also in the digital domain to isolate the components under test, as reported in [Figure 45](#) in which the "Enable" signal is exploited.

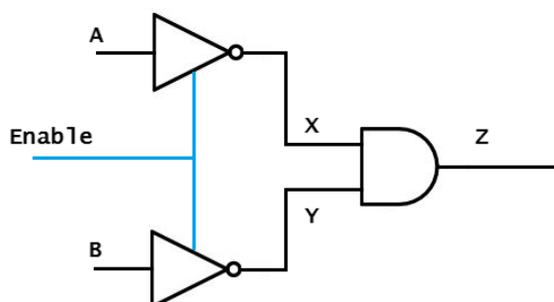


Figure 45: ICT Technique - Digital Test

Analyzing the [Figure 45](#), in order to perform a test to the AND gate without being influenced by surrounding elements, a third driver (in addition to the other used to drive the AND inputs) stimulates the Enable signal of both NOT gates such

that disabling them. The new test pattern sequence, taking into account the digital guarding technique, is reported in [Table 6](#).

Enable	X	Y	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1

Table 6: ICT Technique - Digital Guarding Test patterns

4.2.3 ICT LIMITS

This section describes the ICT limits in terms of defects that this test strategy is not capable of detecting. A panoramic of the most important ICT-undetectable defects and related motivations is reported.

4.2.3.1 Electrically Hidden Components

Electrically hidden components are the ones that result to be not detectable by In-Circuit test due to some electrical limits in the ICT test system. Two categories of them are described:

- **Filter Capacitors:** Filter capacitors are used to filter undesirable frequencies from signals. Very often, different nominal value capacitors (sometimes the difference involves three orders of magnitude) are placed in parallel in order to filter out noises belonging to different frequency ranges. In these configurations, as reported in the [Figure 46](#), the smaller value capacitor can not be covered.

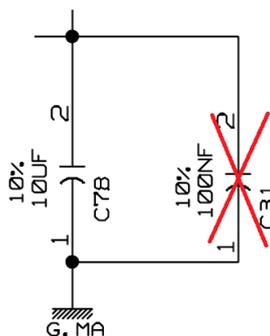


Figure 46: ICT Limit - Filter Capacitor

Analyzing [Figure 46](#), it is possible to identify a capacitors parallel with the respective values of 10uF and 0.1uF (two orders of magnitude difference). In

particular, C31 value (0.1uF) is completely contained into the other capacitor value error which is the 10% of 10uF (1uF). This is the reason why it is not possible to detect the presence or any open pins of C31.

- **RC Parallel:** When a C in parallel to R has to be tested, the current injected to that parallel will charge the capacitor in a time specified by the τ constant³ (Equation 4). If the ATE sensing capability is not enough to get the transient within this period, the capacitor in this parallel can not be detected. An example of RC parallel is reported in Figure 47.

$$\tau \equiv R * C \quad (4)$$

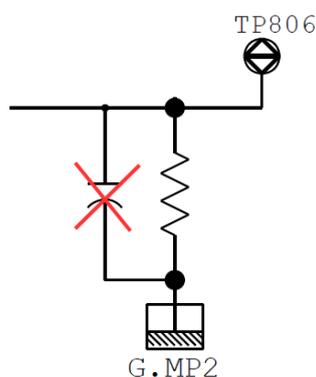


Figure 47: ICT Limit - RC Parallel

4.2.3.2 Diode Scan Limits

"Diode Scan" ICT technique, described in subsection 4.2.2.8, have certain limits. In particular, even though a clamping diode could be detected it is not said that the detection assures that the PIN is soldered well. In fact, if that pin is connected via bus to another clamped pin, two parallel diodes are present and the "Scan Diode" detection does not know which is the diode voltage drop, among the two, which has been measured. To avoid coverage errors, both pins are considered not covered. A clarification example is reported in Figure 48.

³ "It is the time required to charge the capacitor, through the resistor, from an initial charge voltage of zero to approximately 63.2% of the value of an applied DC voltage, or to discharge the capacitor through the same resistor to approximately 36.8% of its initial charge voltage." [33]

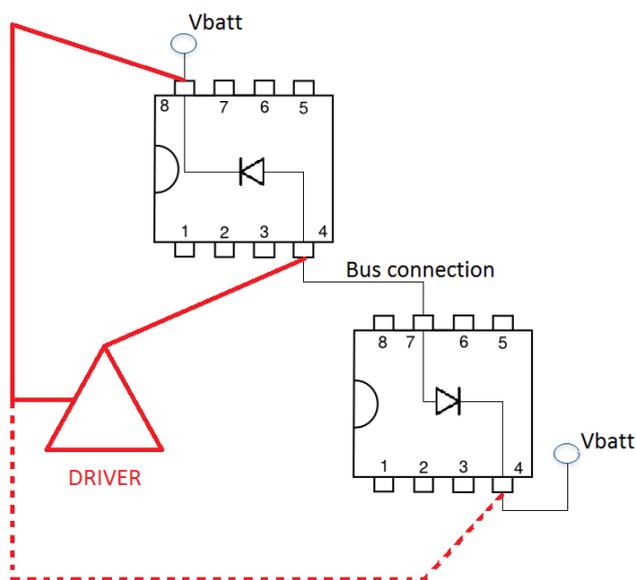


Figure 48: ICT Limit - Clamping Diodes Parallel

In Figure 48, the driver injects a direct current to check if the clamping diode is present between PINS 4-8 on the top IC (clamping to Vbatt). However, the PIN4 of the top IC is connected via BUS to the PIN7 of the bottom IC which in turn is clamped to Vbatt as well. This is a situation of parallel diodes. ICT in this case can not detect void solder faults on both PIN4 of the top IC and PIN7 of the bottom IC.

4.2.3.3 Tied IC pins

Some ICs have multiple GND or VCC pins which are usually tied together. However, there is a limit in recognizing whether any of those pins are affected by an open circuit. This is due to the fact that, among all tied pins, just one is open and others (that are not) still carry the signal not compromising the IC function. This is the reason why the open circuit defect related to these pins can not be detected. An example is shown in the Figure 49.

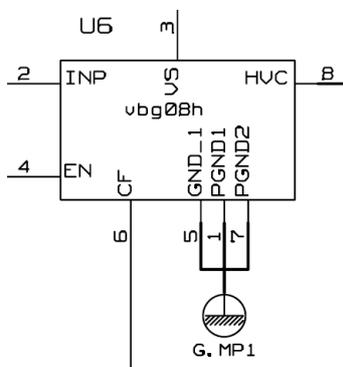


Figure 49: ICT Limit - Tied IC Pins

In this case the open circuit defect of PIN5, PIN1 and PIN7 can not be covered.

4.2.3.4 NC Pins

Several ICs, are often mounted with some purposely not connected (NC) pins. These pins are not soldered to the board and, thus, ICT can not access them. A potential issue could be that adjacent pins (one NC and one soldered) can be wrongly shorted. This can be dangerous for the IC since may be that the NC pin is unexpectedly driven, causing undesired misbehavior.

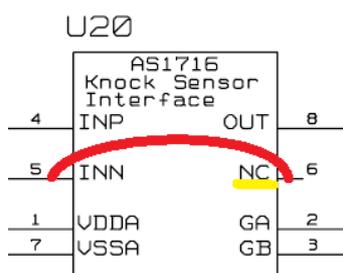


Figure 50: ICT Limit - Shorted NC pins

In the [Figure 50](#), the adjacent pins⁴ PIN5 and PIN6 can be shorted together causing potential misbehavior that ICT can not detect.

4.2.4 ICT DETECTION CAPABILITIES

On the basis of the ICT testing techniques described in the [subsection 4.2.2](#), it is possible to digest the capabilities of In-Circuit Test strategy concerning the detection of the most likely defects summarized in [subsection 3.2.3](#).

It is important to point out that MM-PWT design rules provides that the Test Points are the unique means through which ICT nails can contact the board.

⁴ Apparently, if looking at [Figure 50](#), Pin5 and Pin6 do not seem to be adjacent. However, the figure is taken from a schematic view. By consulting the datasheet of the component it is possible to see that Pin5 and Pin6 are physically adjacent (one close to the other).

<i>ICT</i>	
<i>SOLDER</i>	<i>Short</i>
	<i>Open</i>
	<i>Excess</i>
	<i>Insufficient</i>
<i>COMPONENT</i>	<i>Missed</i>
	<i>Wrong</i>
	<i>Faulty</i>
	<i>Reversed</i>
	<i>Shifted</i>
<i>BGA</i>	<i>Short</i>
	<i>Open</i>

Table 7: ICT - Defects Detection Capabilities

The [Table 7](#) broadly highlights what ICT is capable of detecting in terms of most likely defects.

Concerning **Solder** defects:

- **Short:** Detected by ICT during Pre-Screening phase as described in "Short Test" ([subsection 4.2.2.3](#)) according to the fact that ICT has an access to these pins.
- **Open:** Detected by ICT during Pre-Screening phase as described in "Link Test" ([subsection 4.2.2.2](#)) and also by "Diode Scan" technique ([subsection 4.2.2.8](#)) on IC pins (provided that not other parallel diodes are present, as explained in [subsection 4.2.3.2](#)).
- **Excess:** Undetected by ICT since the electrical behavior of the board is not compromised by this defect. In the cases in which this defect evolves in a short or hides a void, ICT detects these situation as reported in the [Table 7](#).
- **Insufficient:** Undetected by ICT since the electrical behavior of the board is not compromised by this defect. In the cases in which this defect evolves in a void, ICT detects these situation as reported in the [Table 7](#).

Concerning **Component** defects:

- **Missed:** Detected by ICT (it corresponds to have the open defect for any pin of the device).
- **Wrong:** Not always detected since there could be cases in which components are not tested functionally (but only their presence is checked by using the "Diode Scan" technique as described in [subsection 4.2.2.8](#)). ICT detects "wrong" component defect whenever it measures the value of the element (such as resistors, capacitors and inductors).
- **Faulty:** Not always detected since there could be cases in which components are not tested functionally (but only their presence is checked by using the "Diode Scan" technique as described in [subsection 4.2.2.8](#)) as for example multiple-pins IC. ICT detects "faulty" component defect whenever it performs special test on certain components such as operational amplifiers.
- **Reversed:** Not always detected since there could be cases in which components are not tested functionally (but only their presence is checked by using the "Diode Scan" technique as described in [subsection 4.2.2.8](#)) as for example multiple-pins IC. ICT detects "reversed" component defect for those measurable devices that have a polarity such as diodes and electrolytic.
- **Shifted:** Undetected by ICT since the components in this category are considered to be misaligned-mounted but electrically valid. Thus, ICT can not detect this kind of situation.

Concerning **BGA** defects:

- **Short:** Detected by ICT during Pre-Screening phase as described in "Short Test" ([subsection 4.2.2.3](#)) based on the fact that ICT has an access to these pins.
- **Open:** Detected by ICT through the "Diode Scan" technique ([subsection 4.2.2.8](#)) on those pins for which the presence of a clamping diode, not in parallel with other diodes, is ensured (clarification at [subsection 4.2.3.2](#)). Examples of BGA open is a not-soldered BGA pin (which results disconnected from the PCB).

4.2.5 ICT FUTURE

The whole [section 4.2](#) strongly focused on the power of ICT highlighting advantages but also limits on the hardware fault coverage. As it has been described, ICT can not ensure a 100% coverage. In fact, the best strategy line most of the times adopt "ICT + EOL" in order to reach a very high level of coverage.

However, new ideas are coming out such as performing the ICT during the Pick & Place phase of the soldering process ([subsection 3.1.1](#)) consisting in picking the component, measuring its value and placing it on the solder paste before SMD reflow. However, it is very dangerous performing ICT at earlier stage (and so during Pick & Place) since the board is later manipulated during remaining Front-End and Back-End phases ([section 3.1](#)) with very high percentage of being broken up. In this way, all the defects due to manipulation could not be detected anymore since ICT has been already performed. To face these potential problems, more powerful X-ray inspections are being developed in order to help earlier-ICT in terms of aspects (as for example AXI). Note that ICT can not be performed during Back-End since the board is already mechanically sealed and nails could not contact the board anymore. In the other side, EOL (as described in [section 4.3](#)) is a connector-access test and for this reason it is performed during Back-End.

All these ideas are trying to substitute ICT that currently is the test which, if performed singularly, obtains the highest level of coverage with respect to all other techniques even though it requires much more accessibility on the board than for example EOL (which accesses to the connector) or AOI (which do not contact the board at all).

4.3 EOL

"Functional Test is performed as a final manufacturing step (the so called End Of Line - EOL). It provides a PASS/FAIL determination on finished products, before they are shipped to the customers, in terms of validation of the fact that product hardware is free of defects that could, otherwise, adversely affect its correct functioning in a system application" [12].

MM-PWT performs EOL Functional Test during Back-End process (as described in [subsection 3.1.2](#)), when the PCBA has been sealed after having passed the electrical tests. Prior at flushing the application software into the board, EOL validates the product functionality to attest that it is definitely ready to work in field.

In other words, EOL tests:

- PCBA functionality.
- PCBA behaviors in response to precise stimuli.

It is worth noticing that the requirements of the functional test, its development, and procedures vary widely from product to product and system to system [12]. The next sections describe the Functional Test from the MM-PWT point of view.

4.3.1 EOL EQUIPMENT

Generally speaking, the term *equipment* concerns everything that is needed by a strategy to perform the intended test. Due to the EOL complexity, in fact, *equipment* involves a combination of three macro concepts:

- **Firmware:** Described at [subsubsection 4.3.1.1](#).
- **Methodologies:** Described at [subsubsection 4.3.1.2](#).
- **Hardware Equipment:** Described at [subsubsection 4.3.1.3](#).

4.3.1.1 Firmware

The logic heart of an electronic board is nothing but a MCU. The whole board, communicating with the external environment through the connector, receives some input signals which are processed (by some input stages) and presented to the MCU which, in turn, decides what are the actions to be taken. Signals coming out from the MCU are processed by some output stages in order to be presented to the connector (and so to the external environment). This is the operational mode of the board. During EOL, instead, the MCU is not flushed with the application software yet. It is so a virgin IC without a *"brain"*. The Test Firmware has the role to provide to the MCU a certain intelligence in order for the EOL test to be assisted and supported. In this way it is possible to stimulate and read MCU signals from the connector with a precise protocol. During the test, the hardware equipment (ATE) is in charge

of emulating the real environment in order to send and receive signal to/from the DUT with the purpose of ensuring that the DUT is intact in all its functionalities (as reported at [Figure 51](#)).

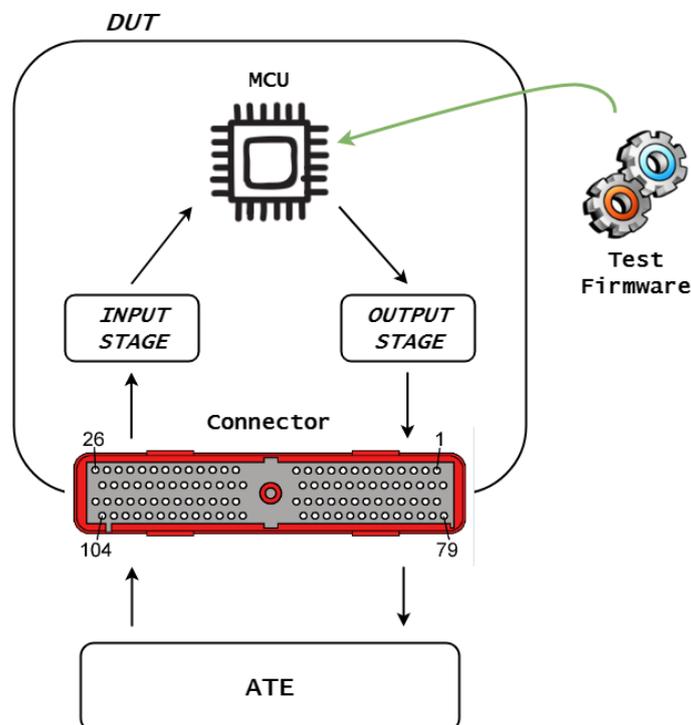


Figure 51: EOL - Test Firmware

4.3.1.2 Methodologies

An important role is played by the methodologies adopted during the test. *What functions of the board are tested? How? Which are the techniques involved?* This is an aspect which varies from company to company and depends on the Testing Team's know-how and past experience. An overview of the MM-PWT EOL methodologies organization is provided in the [subsection 4.3.3](#).

4.3.1.3 Hardware Equipment

The EOL Hardware Equipment is composed of all instruments necessary for the ATE to perform the test. Depends on the philosophy adopted by company, the system equipment can be:

- **Product Dependent:** Instruments composing the ATE are the strict necessary for the test to be performed on a precise board. The equipment, in this case, would fit the board. Costs and timing are kept very low. However, whenever a new board has to be tested, a new equipment has to be built.
- **Generic:** A generic ATE is assembled for a family of products. The advantage, in this case, is that it can be used for a huge variety of products. The drawback

is related to the excessive costs to compose the ATE. It is important to point out that the ATE hardware constitutes an emulation of the hypothetical real environment in which the DUT should work. This environment is the union of instruments (within the ATE) and Loads (connected to the ATE). The latter ones are usually *specific* for the board under test. When the ATE-Generic approach is adopted, it is better to have the possibility to connect and disconnect them to the ATE. In this way it is possible to use the same equipment for slightly different products by changing only the loads.

4.3.1.4 Ideal EOL Architecture

The three aspects highlighted in the previous sections (Firmware, Methodologies and Hardware Equipment) are three concepts strictly related and they have to go hand by hand. It does not make any sense to have the firmware which is capable of reading parallel inputs if the channel connecting the ATE to the DUT is only one. Or more, it is useless having plenty of channels and a firmware which is capable of operating in parallel, if no parallel tests are provided by the methodology book.

From the above examples it is immediate to get that all aspects have to be improved in parallel, otherwise the worst one creates a bottleneck in the capability exploited by the test, as reported in [Figure 52](#).

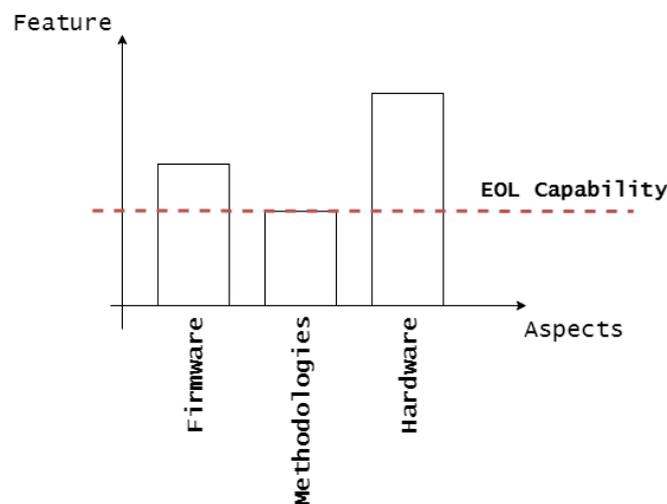


Figure 52: EOL - Architecture Capability

4.3.2 EOL TECHNIQUES

Differently than ICT, in which structure of the test has a prefixed structure, the MM-PWT Test Engineer, while developing the functional test, interacts with four dependent entities:

- **Methodology Book:** It constitutes a kind of "*Lesson Learn*" for the whole team. It is a reference point continuously updated which describes in a generic way

(independent from the adopted test equipment) how to perform certain tests. It is very difficult to generally explain those concepts, but the role of the methodology book is to *instruct* the Test Engineer and give him an idea of what has to be done to test certain situation.

- **Prescription:** It is a specification involving technical details of all instruments needed for the test to be properly performed (what you need from ATE, what you need in terms of load to simulate the external environment, what you need in terms of firmware and so on). It is created by the Test Engineer that, after a study of the electrical schematic to come up with the tests to be performed (generically reported in the Methodology Book), defines all specific details that are involved in these tests.
- **File Sequence:** It is the sequence of all tests applied to the board during EOL, divided in four main operational modes: *DUT_OFF*, *DUT_ON* and *DUT_EOL*. The software used to describe the sequence (TestStand) creates a layer of routine calls. In turn, the routines are real device drivers for the ATE hardware instruments. In this way, the Test Engineer can control the ATE instruments to stimulate or sense the board in a simpler manner, as shown in [Figure 53](#). The Panel is the hardware interface for the DUT to communicate with the ATE instruments.

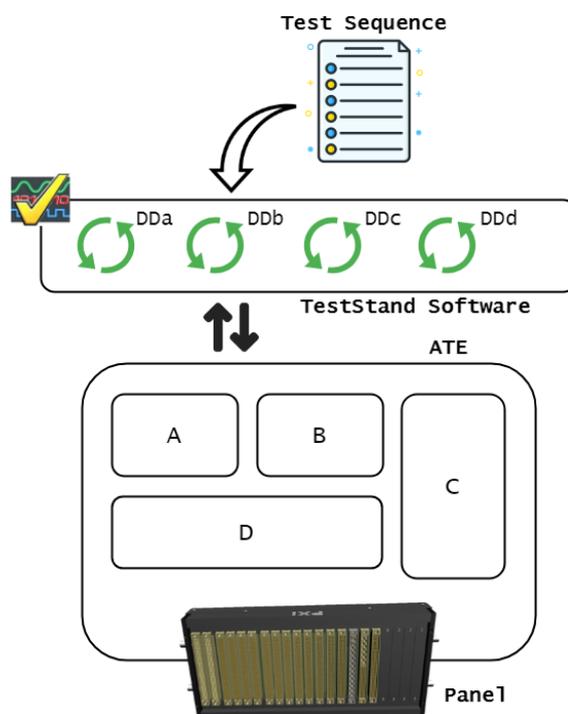


Figure 53: EOL - TestStand Layer

- **Datalog:** It is the output document produced by the ATE at the end of the performed functional test, reporting technical details about the test steps and

whether they have PASSED or FAILED. This document is very important during the mass production phase since it is interpreted by the operator in order to decree whether the product has to be discarded (in case of FAIL) or sent to following phases in the production flow (in case of PASS).

The common flow followed by the Test Engineer is the following one:

1. **ECU Study:** A first analysis is performed on the Electrical Schematic to understand what are the main functions implemented by the board and start coming up how to test them.
2. **Prescription Drafting:** Once that the idea of what tests to be performed is consolidated, the prescription is drafted since the Test Engineer is now aware of most of the technical details involved (such as voltage and current involved, ranges to be compared with read values, characteristics of the instruments used and so on).
3. **Test Sequence Drafting:** After having defined the technical details of the test steps to be performed, the real test sequence is created. The DUT is tested in different operating modes:
 - *DUT_OFF*: In this mode, the DUT is not power supplied. The ATE, communicating with the connector, can perform an electrical check (link, open and short tests) on the connector pins only. Stages, as well as the MCU pins, can not be tested since the MCU is off and so the firmware can not run in support to the test, as shown in [Figure 54](#)

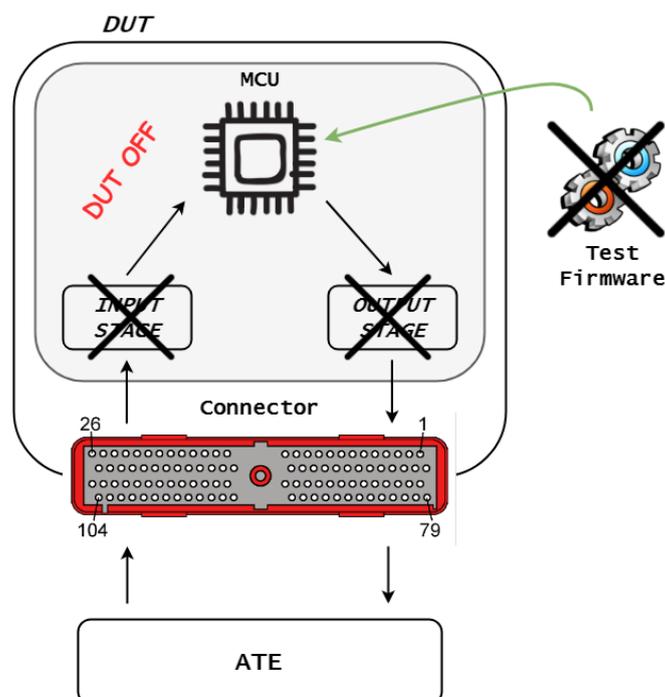


Figure 54: EOL - DUT_OFF Test

- *DUT_ON*: In this mode, the DUT is power supplied. Now stages can be electrically checked since a closed loop between ATE and MCU (now ON) is created. The added value of this mode is given by the firmware that can ease the testing activity.
- *DUT_EOL*: In this mode, the DUT is power supplied and tested in its functionalities. Basically, the conditions are similar to the *DUT_ON* mode with the difference that the test is not only electrical but also functional. The functions of the board, performed by a group of components, are tested.
- **Test Sequence Debugging**: Together with the Test Sequence Drafting, a debug phase is performed. It consists in tuning the test sequence such that making it stable more and more till reaching the final version.

4.3.3 EOL LIMITS

The EOL limits are mostly related to the precision in the measurements of the component value. The following description is referred to a Resistor, but it can be extended to any kind of component. The measurement principle to test a resistor is the same as the one used by ICT ([subsection 4.2.2.4](#)). However, ICT can directly access to the resistor pins (as far as the test points are present) whereas EOL follows a path that starts and ends from/to the connector pins traversing the input/output stages and the MCU which operates according to the firmware directives. In addition to this path (whose impedance is known and computable from the schematic), an additional critical impedance (whose value is uncertain due to deterioration and impossibility to be reconstructed) is introduced by the ATE in the complex interconnection wiring system between instruments and panel, as reported in [Figure 55](#).

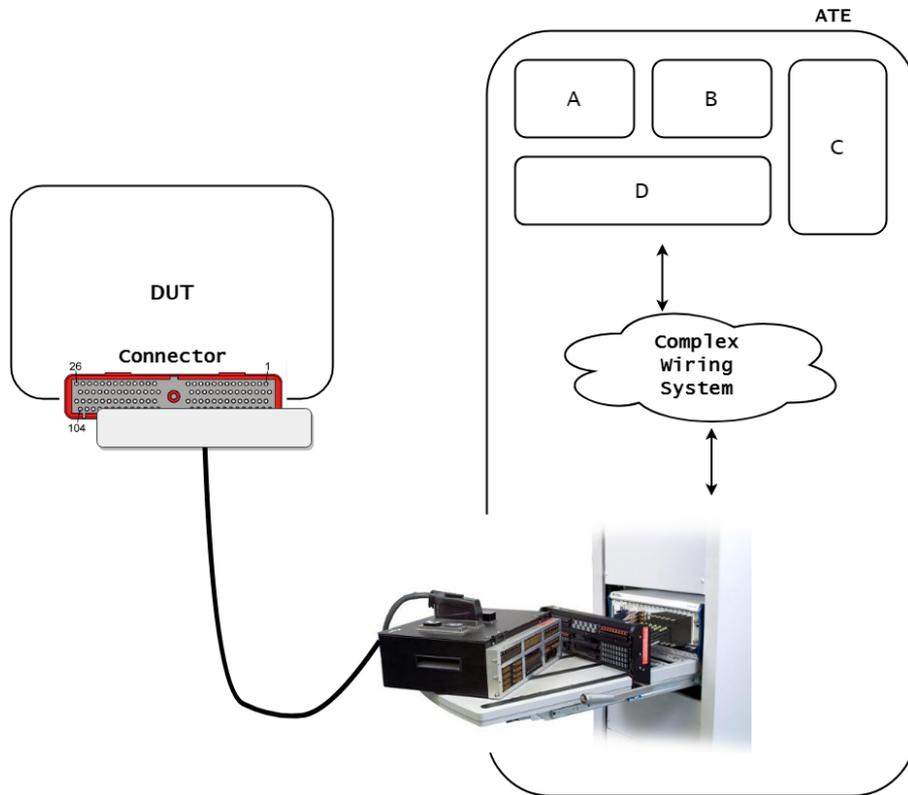


Figure 55: EOL - ATE Instruments/Panel Impedance

In [Figure 56](#), it is described the reason why this impedance uncertainty degrades the quality of the test. The line in the middle represents the nominal value of the component (for instance a Resistor *ohmic* value). The component, in turn, always presents some tolerance errors due to the production process. Thus, the green area represents the expected value range. Ideally, the test returns PASS if the ATE read value is within the green area, FAIL otherwise. Practically, also the uncertainty introduced by the ATE-DUT wiring has to be considered and so the PTol and NTol limits can vary. With the incoming of the regulations concerning safety-critical components (ISO 26262), it is safer considering the yellow range. However, whenever a component is characterized by having a value within the dark-green area, the test returns FAIL and the board is discarded even though it is good. This is a non-negligible drawback but it is a good compromise to accomplish safety regulations.

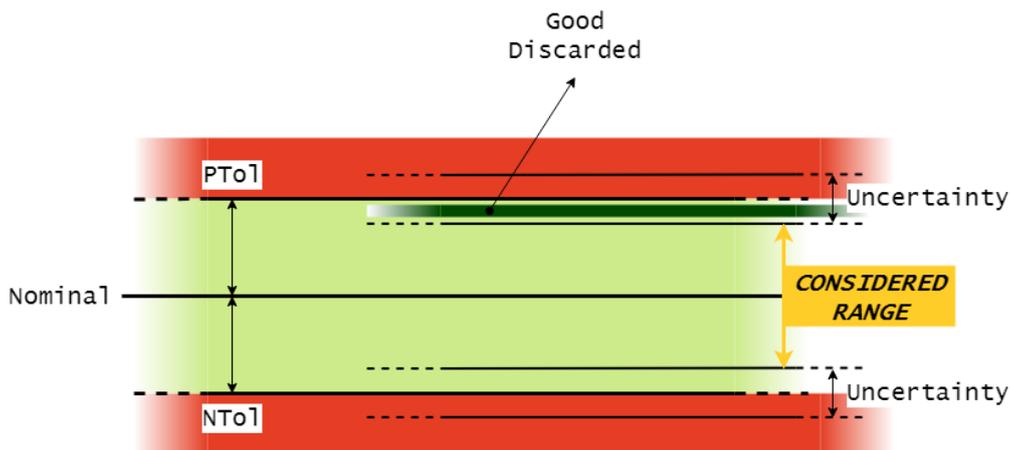


Figure 56: EOL - Safety Tolerance Ranges

4.3.4 EOL DETECTION CAPABILITIES

On the basis of the EOL testing feature described in the subsection 4.3.3, it is possible to digest the capabilities of the Functional Test strategy (EOL) concerning the detection of the most likely defects summarized in subsection 3.2.3.

It is important to point out that the real added value provided by EOL is the connector pins full-coverage (not covered neither by ICT nor AOI).

<i>EOL</i>	
<i>SOLDER</i>	<i>Short</i>
	<i>Open</i>
<i>COMPONENT</i>	<i>Excess</i>
	<i>Insufficient</i>
	<i>Missed</i>
	<i>Wrong</i>
<i>BGA</i>	<i>Faulty</i>
	<i>Reversed</i>
	<i>Shifted</i>
<i>BGA</i>	<i>Short</i>
	<i>Open</i>

Table 8: EOL - Defects Detection Capabilities

The Table 8 broadly highlights what EOL is capable of detecting in terms of most likely defects.

Concerning **Solder** defects:

- **Short, Open:** Not always detected since some components in the stages, even though shorted or absent, leave some functions of the board intact (for instance, some capacitors are placed to filter some noises. If the noise is not present, the capacitor absent can not be detected).
- **Excess:** Undetected by EOL since the electrical behavior of the input and output stages is not compromised by this defect.
- **Insufficient:** Undetected by ICT since the electrical behavior of the input and output stages is not compromised by this defect.

Concerning **Component** defects:

- **Missed:** Detected by EOL (a missed component would compromise some functions of the board).
- **Wrong:** Detected by EOL (a wrong component would compromise some functions of the board).
- **Faulty:** Detected by EOL (a faulty component would compromise some functions of the board).
- **Reversed:** Detected by EOL (a reversed component would compromise some functions of the board).
- **Shifted:** Undetected by EOL since the components in this category are considered to be misaligned-mounted but electrically valid (the stage is left intact). Thus, EOL can not detect this kind of situation.

Concerning **BGA** defects, they are all detected by EOL since they would compromise electrical aspects of the MCU and so, in turn, some functions of the board.

4.3.5 EOL FUTURE

The unpredictability of EOL recently gave birth to the desire of having an higher modularity in Functional Test. The idea behind this need is to pass from a proprietary functional tester to a modular test system that can be reconfigured. In this way, functional test would become a library of different test routines. Thus, a routine is invoked to test a particular component whereas the functional test parameters for that precise device would be provided by the component silicon vendor (for instance, the routine to test a resistor would be the same. Instead, the parameters would change in relation to the resistor characteristics).

In other words, it would mean moving from the functional test hardware-approach (in which the function is declared "PASS" if and only if all hardware components involved singularly work as expected) to a software-based implementation (in which the function is declared "PASS" if the produced functional results are the expected ones). Very probably, migrating from the hardware approach to a software-based one, would neglect some situation visible only at lowest granularity and make more difficult the possibility of performing diagnosis.

Part III

HARDWARE FAULT COVERAGE

TEST COVERAGE

Test coverage is a discipline which studies the quality of the test line acting during production (described in the [chapter 4](#)). It has become more and more a key qualification tool for driving schematic design, board layout and test program development in order to achieve the best production yield.

Next sections introduce the basic coverage theory concepts ([section 5.1](#)). After that, a description of the the traditional metrics is reported ([section 5.2](#)) in order to highlight the main aspects concerning the Defect Universe (target of the test - [section 5.3](#)). Finally, the coverage document contents are presented ([section 5.4](#)).

5.1 Coverage Introduction

The term *TEST COVERAGE* is essentially an indicator resuming the test strategy line capability of detecting all potential defects that can arise during manufacturing. A *DEFECT* is considered *COVERED* if and only if its eventual presence is detected by the test line. On the other side, a *DEFECT* is considered *UNCOVERED* if the test line is not capable of detecting it anyway.

The higher number of *covered* defects, the higher the quality of the test line (*coverage*), the lower the number of bad products sold to the customer.

The concepts described above seems to be quite rational and intuitive but actually they are not. In fact, considering all defects that can arise during manufacturing is not practically possible (since they tend to an infinite number). For this reason the test target is based on a theoretical model which tries to list as many defects as possible (this set of defects is called "defect universe", described at [section 5.3](#)).

In [Figure 57](#) is reported the test coverage flow. The choice of a metric applied on a product determines the defect universe. Once that the list of all defects has been defined, a successive phase provides to compile the coverage document by annotating from which tests (if any) the defects are covered. After that, the final Test Coverage percentage can be computed either weighted (each defect has a precise weight causing different defects to impact more or less on the final coverage) or averaged (all defects have same weight and so an equal impact on the coverage).

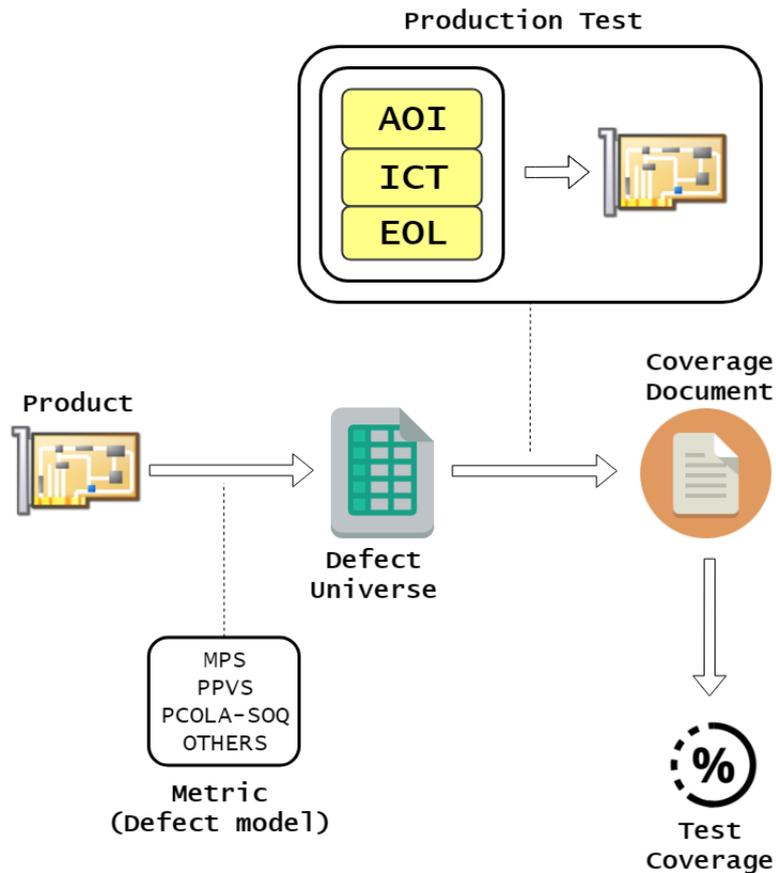


Figure 57: Test Coverage Flow

Next sections describe the steps involved in this process

5.2 Coverage Metrics

As described in [chapter 3](#), the manufacturing process is not error-free due to machinery and process imprecision. This is the reason why the test discipline plays an important role and motivation to exist. The choice of a metric, to categorize all possible defects, is needed to understand what are the hardware faults to be covered during the test (it is a real test target definition) and also to give a meaningful description to the final "Test Coverage" percentage in terms of which defects have been really covered and which not (only a percentage would be a meaningless result).

Next subsections explain different kind of metrics (MPS, PPVS, PCOLA/SOQ) through which the defect universe is built up. These are the results of a standard process aiming to unify the concept of board test coverage. However, each company is also free to implement a proprietary coverage based on its own metric (as explained in [section 6.1](#) in the case of Magneti Marelli Powertrain).

5.2.1 MPS

The MPS (which stands for Material, Placement and Solder) is a metric that defines the following manufacturing defect categories:

- **Material** : It is related to the supply material phase. It models the correctness of the supplied component.
"Is it the right component?"
- **Placement** : It is related to the component placement phase. It models the correctness of the component-placement.
"Is the component rightly placed on board?"
- **Solder** : It is related to the component soldering phase. It models the correctness of the component pins solder (the component solder attribute summarizes solder defects related to all pins of the device).
"Are the component pins rightly soldered on board?"

It is the highest-level representation of the production flow since the defects refer to the three macro stages of the line: Supply of material, Placement and Solder of components. However, the representation of this manufacturing defects can be further broken down into hierarchical blocks, as reported in [Figure 58](#).

5.2.2 PPVS

The PPVS (which stands for Presence, Polarity, Value and Solder) is a metric that defines the following manufacturing defect categories:

- **Presence** : It is related to the component placement phase. It models the presence of the component after the assembly process.
"Is the component present on board?"
- **Polarity** : It is related to the component placement phase. It models the correctness of the component-placement in terms of polarity (which refers to the "Wrong polarization" defect explained in [subsection 3.2.2](#)).
"Is the component rightly reversed on board?"
- **Value** : It is related to the supply material phase. It models the correctness of the mounted component.
"Is it the right component?"
- **Solder** : It is related to the component soldering phase. It models the correctness of the component pins solder (the component solder attribute summarizes solder defects related to all pins of the device).
"Are the component pins rightly soldered on board?"

PPVS is an example of attribute-depth. In fact, MPS "Placement" is, in PPVS, broken down into "Presence" and "Polarity" (as reported in Figure 58). This is reasonable since a placement error can involve several defects such as missing-placement (Presence) and reversed-placement (Polarity). However, the PPVS metric is still far from the reality since a high number of defects are still not modeled. For instance, concerning "Solder" property, what does it mean "rightly soldered"? Is it referring to the quantity of solder material? Or is it referring to an absent connection between component and board? Or even is it referring to a potential short? A further defect decomposition is needed in order for the resulting DU to provide a precise set of defects closer to the real one.

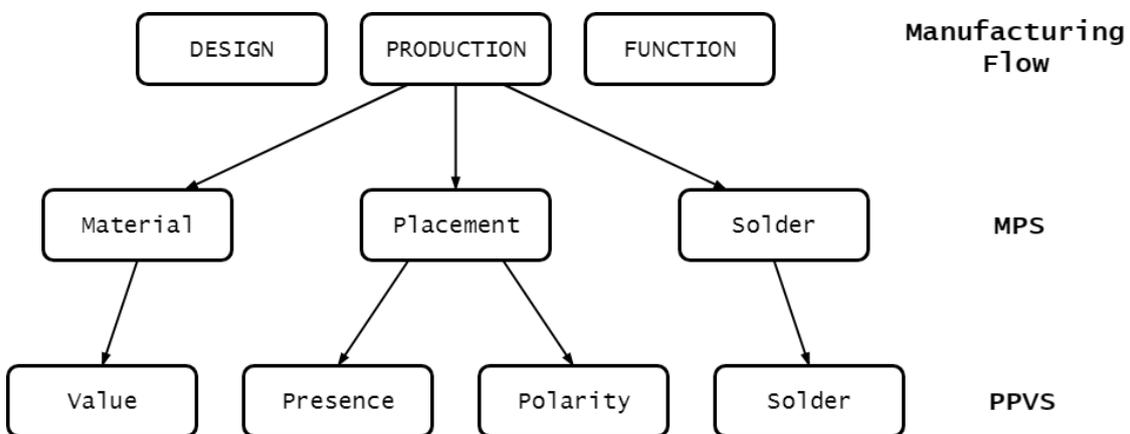


Figure 58: Test Coverage - MPS and PPVS metrics

5.2.3 PCOLA/SOQ

The PCOLA/SOQ (which stands for Presence, Correct, Orientation, Live, Alignment, Short, Open and Quality) is a metric that defines two broad categories of manufacturing defects.

The first group, involving component properties, is characterized by the following defects:

- **Presence** : It is related to the component placement phase. It models the presence of the component after the assembly process.
"Is the component present on board?"
- **Correct** : It is related to the supply material phase and significant only if the component is present. It models the correctness of the mounted component.
"Is the mounted component the right one?"

- **Orientation** : It is related to the component placement phase. It models the correctness of the component-placement in terms of polarity (which refers to the "Wrong polarization" defect explained in [subsection 3.2.2](#)).
"Is the component rightly reversed on board?"
- **Live** : It is related to the component operation phase. It models the fact that the device is able to work or not. Note that it is not a full functional qualification but it refers to the *aliveness* of the component.
"Is the component working (no matter how)?"
- **Alignment** : It is related to the component placement phase. It models the correctness of the component-placement in terms of shifting (which refers to the "Shifted Components" defect explained in [subsection 3.2.2](#)).
"Is the component rightly aligned on board?"

It is important to notice that Presence, Correct, Orientation and Live attributes are fundamental for the performance of the device, whereas the last (Alignment) is purely a qualitative property [24].

The second group, concerning connection¹ properties, is characterized by the following defects:

- **Short** : It is related to the soldering phase. It models the presence of an undesired electrical continuity to other connection points on the board.
"Is the connection point undesirably shorted to other connection points?"
- **Open** : It is related to the soldering phase. It models the presence of an undesired lack of continuity between the connection point and the board.
"Is the connection point undesirably not soldered to the board?"
- **Quality** : It is related to the soldering phase. It models the quality of the solder associated to the connection point.
"Is the connection point characterized by a good quality solder?"

Note that Short and Open attributes are fundamental for the performance of the board, whereas the last (Quality) is purely a qualitative property [24]. The component solder attributes (Short, Open and Quality) summarizes solder defects related to all pins of the device (Short, Open and Quality are considered for each pin), as shown in [Table 9](#).

¹ A connection is a place where the component is electrically attached to the board through solder material (a capacitor has two connections, an IC could have up to hundreds connections).

PCOLA/SOQ Component	Pin	SOQ
Presence	Pin1	Short
Correct	Pin1	Open
Live	Pin1	Quality
Live	Pin2	Short
Alignment	Pin2	Open
Short	Pin2	Quality
Open		
Quality		

Table 9: PCOLA/SOQ - Connection Properties

In this case the left table represents the defect-set of a two pins component. The PCOLA attributes are component-dependent whereas the SOC defects are the summary of the Short, Open and Quality referred to the pins of the component (table on the right).

5.2.4 CONCLUSION

It is now useful summarizing the characteristics coming from the various metrics in order to define which is the best one regarding on the company needs. The [Table 10](#) gives a panoramic of the granularity of MPS, PPVS and PCOLA/SOQ:

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

Table 10: Metrics - MPS, PPVS and PCOLA/SOQ relations

It can be noticed that PPVS involves a MPS specialization in terms of Placement (Presence + Polarity). Instead, PCOLA/SOQ fully specialize MPS:

- **Material** : It is broken down in Correct and Live. When a component is supplied it can be the wrong one (Correct) or it can be the right one but not able to work (Live).

- **Placement** : It is broken down in Presence, Alignment and Orientation. A placed component can be wrongly reversed (Orientation), shifted with respect to the expected alignment (Alignment) or not present at all (Presence).
- **Solder** : It is broken down in Short, Open and Quality. A connection between PCB and component can be shorted to another solder point (Short), can be void (Open) and can be characterized by a bad quality (Quality).

It is immediate to state that **PCOLA/SOQ is the most rigorous technique which characterizes the potential defects affecting a single component and its pins**. However, still some real defects are missing in this model (as for example the functional ones). There is in fact a new iNEMI's FAIM extension² to PCOLA/SOQ which is being recently used. To conclude, noticing the similarity to the potential defects summary reported at [subsection 3.2.3](#), it can be said that PCOLA/SOQ is a very good compromise for defect modeling and, due to its completeness, some properties should be even ignored when computing the final coverage (as for example "Orientation" for resistors which, even if reversed, they still work as expected). All these properties makes **PCOLA/SOQ a standard metric suitable for MM-PWT purposes**.

5.3 Defect Universe

The Defect Universe represents a subset of all real defects from which a board can be affected to. It is based on a metric which, as reported in [section 5.2](#), represents a "Defect Model" for a single component. Thus, the defect universe is the list of all defects concerning all components of the board.

The DU has a high importance since it is both a test requirement (in terms of how many defects should be covered from the test line) and a factor strongly influencing the precision of the "coverage" as test quality indicator. Some following examples clarify the concept of defect universe.

The following list states the colors meaning, concerning [Figure 59](#) and [Figure 60](#):

- **Blue** : It involves the set of all possible real defects from which a product can be affected to.
- **Red** : It involves the set of the modeled defects of the product. This is the DU (target of the test line in terms of coverage).
- **Green** : It involves the set of all real defects that the test line covers.

² It is the PCOLA/SOQ/FAIM in which functional defects such as Feature, At-speed, In Parallel, Measurement are involved. [5]

- **Orange / Red** : The percentage of red area covered by the orange represents the subset of the DU defects which are covered by the test line. This is the test coverage.

In relation to [Figure 59](#), in which the DU models much less defects than real (since the red area covers a small part of blue), the test seems to be capable of covering most defects of the board (and so test coverage percentage is high) but actually lot of real defects are uncovered (blue area minus green one). The problem here is that, due to the weakness of the test requirements, the test line relaxes and no effort is added by test engineers to improve its quality. This is a proof that a not precise Defect Universe makes the test coverage a bad indicator for the test line quality.

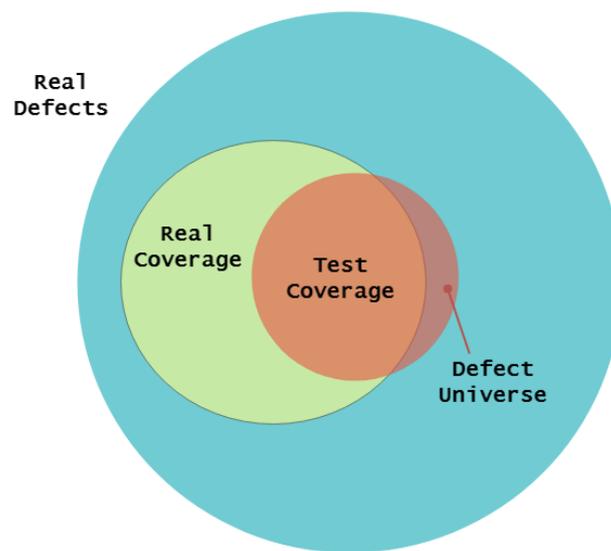


Figure 59: Defect Universe - Far from reality

In relation to [Figure 60](#), where the defect universe is closer to the reality (since the red area covers a consistent part of blue), more DU defects are uncovered with respect to the previous case causing the coverage to assume lower percentage values. This is a case in which a more precise DU makes the Test Coverage a good indicator of "test line quality".

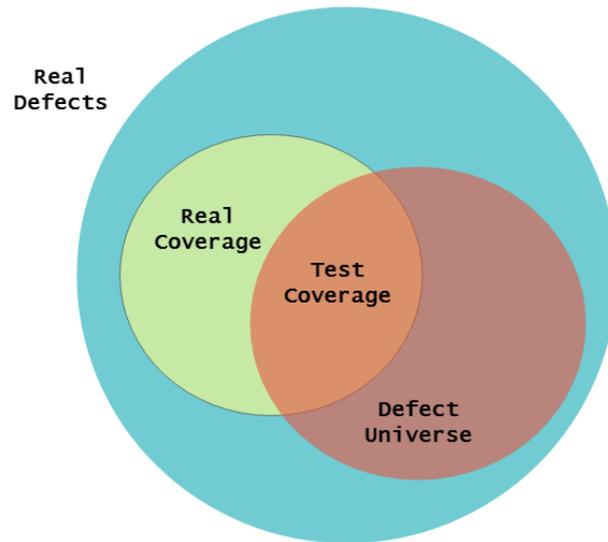


Figure 60: Defect Universe - Close to reality

To conclude, it can be said that the more DU fits the reality (red area closest to the blue one), the more the test coverage is a good indicator for the quality of the test line. This is the reason why it is important to choose the best metric from which the DU is built up. A question which makes reasoning about these concepts is: *"Is a board 'good' because the test passes?"*. The answer is probably "yes" but only if test requirements (metric and defect universe) are consistent with the reality [9].

5.4 Coverage Document

This section reports the flow adopted to generate the coverage document.

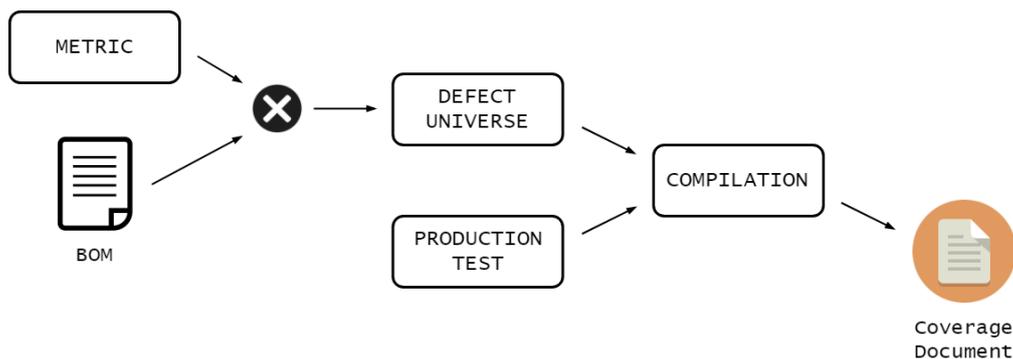


Figure 61: Test Coverage Document - Flow

The [Figure 61](#) summarize the following process:

1. **Metric Selection** : In this phase the metric to characterize potential defects for any component is chosen. In [section 5.3](#) it has been analyzed the importance of the DU. It is the reason why the choice of the metric is a crucial step before moving forward.

2. **Defect Universe Generation** : Once that metric has been chosen, each component is theoretically modeled with a certain set of defects. Thus, in this phase, a Cartesian product between the list of components (BOM) and list of defects per component (metric model) is performed such that obtaining the full list of defects that the whole board can be affected to. The output of this phase is nothing but a coverage document which has not been compiled yet. Note that: A netlist can be used instead of the BOM to collapse equivalent defects. This is allowed by the fact that netlist includes pin connections info and so defects related to pins belonging to the same net can be collapsed. Analysis like these are reasonable for ICT test (in which an electrical check is performed), but not for AOI (in which the two solders, even if connected to the same net, are considered independent). The goal of the test coverage is to best represent the defectiveness of the board and, based on that, attest which is the quality of the line. This is the reason why these steps (as the defect universe generation) must remain as much independent as possible from the adopted test strategy line. Thus, the BOM is used to produce the DU. [24].
3. **Coverage Document Compilation** : The empty coverage document (defect universe) is now compiled by annotating what defects are covered from any test. The idea behind the compilation is the following:

*TEST Y covers DEFECT X
if and only if
Y notices an unwanted situation when X occurs.*

An example is the short defect between two pins of a big resistor. It is covered by ICT since, if the defect occurs, ICT notices this situation (performing a "Short Test" as reported in [subsubsection 4.2.2.3](#)). Another example is the absent defect of a filter capacitor. It is uncovered by ICT since, if the defect occurs, ICT does not notice this situation (even though performing the "Capacitor Test" as reported in [subsubsection 4.2.2.5](#)).

Based on these examples, the [Table 11](#) reports the result of the compilation applied to the circuit portions reported in [Figure 62](#).

Component	Defect	AOI	ICT	EOL
...
R137	Short			
...
C32	Presence			
...

Table 11: Coverage Document Compilation - Document Example

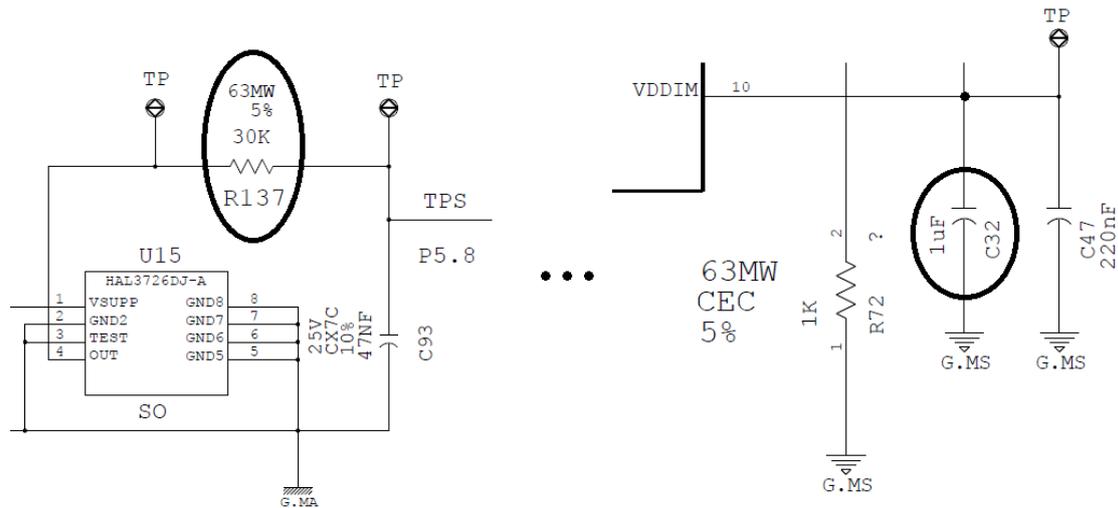


Figure 62: Coverage Document Compilation - Circuit Samples

5.5 Final Test Coverage

The information reported within the coverage document are used to compute the final coverage as percentage of board defectiveness potentially detectable by the test line. As anticipated, the coverage can be computed in two different ways:

- **Averaged Coverage:** In this case the coverage is simply computed as reported in Equation 5. Defects have the same weights. They equally impact on the coverage.

$$COVERAGE_{averaged} \equiv \frac{\#Defects_{covered}}{\#Defects} \quad (5)$$

- **Weighted Coverage:** Considering all defects with same weights is not reasonable since they have different probability to arise. In the automotive fields, where requirements from customers are very strict (usually production should guarantee 1 DPMO³), each defect is weighted with its own probability to arise. This is reasonable since it is more serious not covering a defect which has high probability to occur with respect to a one which hardly ever arises. The higher

³ DPMO, standing for "Defect Per Million Opportunities", represents the opportunity of a component to present a certain defect. In order to better understand this concept let's suppose to consider a resistor which is characterized by 10 DPMO for its "Presence" defect: this means that 10 out of one million times this resistor will not be mounted at all. Component DPMOs are usually results of studies or they can also be taken from standard databases (as for example Siemens SN29500).

the DPMO of not covered components, the lower the coverage. The weighted coverage formula is reported in Equation 6.

$$COVERAGE_{weighted} \equiv \frac{\sum DPMO_{covered}}{\sum DPMO} \tag{6}$$

An example of coverage computation is reported in Table 12. For any defect, the DPMO column defines the probability of a defect to occur. The green cells indicate which defects are covered by the whole test line for any of the three 2-pins components (R2, C4 and L1).

	Device DPMO	R2	C4	L1	Defects	Covered Defects	DPMO	Covered DPMO
Presence	10			1	3	1	30	10
Correct Orientation	1		1		3	1	3	1
Live	4				3	0	12	0
Alignment	25				3	0	75	0
	100	1		1	3	2	300	200
1 Short	30		1		3	1	90	30
1 Open	20	1			3	1	60	20
1 Quality	5		1		3	1	15	5
2 Short	30	1	1	1	3	3	90	90
2 Open	20			1	3	1	60	20
2 Quality	5	1			3	1	15	5
TOTAL					33	12	750	381

Table 12: Test Coverage Percentage - Computation Example

The two difference coverage computations are:

$$COVERAGE_{averaged} \equiv \frac{12}{33} \equiv 36.36\% \tag{7}$$

$$COVERAGE_{weighted} \equiv \frac{381}{750} \equiv 50.8\% \tag{8}$$

The weighted coverage results higher than the averaged one due to the fact that the most critical defects (the ones with highest DPMO) are mostly covered (Short and Alignment in particular).

MM-PWT TEST COVERAGE

In [section 2.1](#) the MM-PWT product has been introduced. After that, the production line has been described ([section 3.1](#)) in order to focus on the test strategies impacting on the hardware fault coverage ([chapter 4](#)). At this point, a full characterization of the MM-PWT test coverage implementation has to be provided. *Which is the adopted metric? How the coverage document is built? How the final coverage is computed?* This chapter answers to these questions describing the current MM-PWT coverage aspects aiming to highlight the main limits and motivation to migrate towards a different flow.

6.1 Proprietary Metric

In [section 5.2](#), a panoramic of the traditional metrics has been provided. The responsibility of a metric is to provide a set of potential defects for each component. MM-PWT adopted a proprietary metric ($\text{MM-PWT}_{\text{metric}}$) whose characteristics are reported below:

- **Adaptive:** The metric is adaptive in the sense that the model of defects is not unique but it "adapts" to each precise component. In other words, a traditional metric, as for example PPVS, identifies Presence, Polarity, Value and Solder defects for each component regardless its geometry, functionality or pin configuration whereas the $\text{MM-PWT}_{\text{metric}}$ defines a component-specific defect model. The difference PPVS/ $\text{MM-PWT}_{\text{metric}}$ is reported in [Table 13](#) (a 3-pins transistor has been considered - [Figure 63](#)).

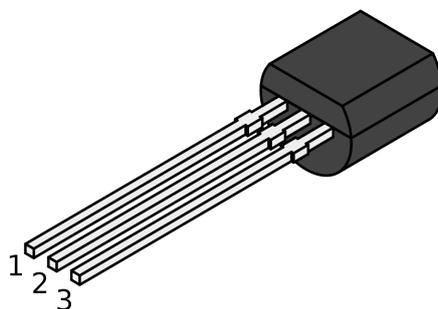


Figure 63: Transistor (3-pins)

PPVS	MM-PWT
Presence	Presence
Polarity	Polarity
Value	Alignment
Solder 1	Open 1
Solder 2	Open 2
Solder 3	Open 3
	Short 1-2
	Short 2-3
	Quality 1
	Quality 2
	Quality 3

Table 13: MM-PWT_{metric} - Adaptivity

In addition to the presence and polarity which are taken into consideration in both metrics, MM-PWT_{metric} models the wrong alignment, the open circuit in all pins, the short between adjacent pins (in fact Short 1-3 is not taken into consideration) and the quality of the solders. The parameter which is considered by PPVS but is missing in MM-PWT_{metric} is "Value", which is modeled only for those components with precise nominal values and silicon-manufacturing tolerances (such as Resistors, Capacitors and so on).

- **Dynamic:** MM-PWT_{metric} is a dynamic metric because it requires to be continuously upgraded. Whenever a new component is encountered, an associated defect model has to be created. Differently, traditional metrics are static in the sense that a fixed number of attributes is defined and never changes.
- **Manual:** New component defect models are created manually. There is no an automatic process which generates a defect model based on a component.
- **Error-Prone:** Manual updates to the model cause MM-PWT_{metric} to be strongly subjected to human errors.
- **Rules-Compliant:** Even though MM-PWT_{metric} is an adaptive model, it follows some rules to build up the fault model for each component:
 - "*Quality*" solder property is modeled for each pin.
 - "*Presence*" and "*Alignment*" are modeled for any component.
 - "*Short*" is modeled for adjacent pins.
 - "*Value*" is modeled for the measurable components.
 - "*Polarity*" is modeled for those component which have a reversibility.

- **Storage-dependent:** MM-PWT stores the component models into a file ("*FaultModel.ini*"). Whenever a new model has to be created, it is appended to the file. In this way, *FaultModel.ini* contains the models of all previously encountered components. The file is processed by a script during coverage document generation (as described in [subsection 6.2.1](#)). The file content involves a list of components with associated *Part Number*¹ and *Defect Model* (as reported in [Table 14](#)).

Part Number	Fault Model
NF.0107044.A	Presence
	Alignment
	Open 1
	Open 2
	Open 3
	Short 1-2
	Short 2-3
	Quality 1
	Quality 2
	Quality 3
...	...
ND.0023566.A	Presence
	Alignment
	Short 1-2
	Quality 1
	Quality 2
...	...

Table 14: MM-PWT_{metric} - *FaultModel.ini* structure

Note that, within "*FaultModel.ini*", the fault models definition, reported in [Table 14](#), must follow a particular syntax in order for the file to be properly processed during the coverage document creation ([subsection 6.2.1](#)).

¹ "It is an identifier of a particular part design used in a particular industry. Its purpose is to simplify reference to that part." [30] In Magneti Marelli Powertrain, a part number identifies a component which is unique in terms of characteristics, geometry and functionalities, no matter the supplier company which produces it. In fact, if the component is supplied by a different company, it would be still identified with the same part number.

6.2 Proprietary Coverage Document

This section reports a well-rounded description of the Coverage Document developed by Magneti Marelli Powertrain. This document, after deep studies, has been chosen as target of improvement in the Powertrain coverage field.

6.2.1 CREATION

Pre-conditions:

- **Component List:** *BOM* file (full list of the board components).
- **Fault Model:** *FaultModel.ini* file specifying defect model for each component within the BOM.

Post-conditions:

- **Empty Coverage Document:** Empty document ready to be compiled.

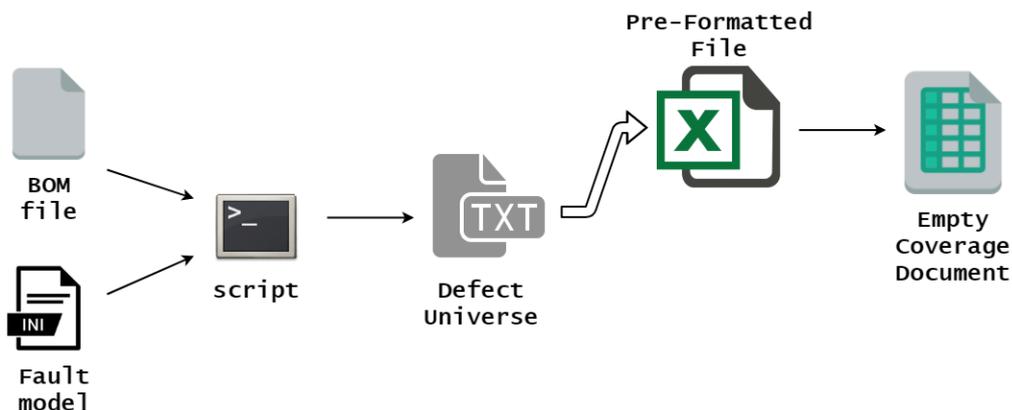


Figure 64: MM-PWT_{coverage} – Coverage Document Creation Process

The creation flow, resumed at [Figure 64](#), requires to launch a cross-elaboration script (executable file) which makes an *inner join*² between BOM and Fault Model files such that creating a list of potential defects (grouped by component) that can affect the board (defect universe in a "txt" format). The list of defects is then imported into a virgin pre-programmed Excel file (it involves a pre-set format with formulae and styles ready to be compiled for an auto final coverage generation). The resulting output of the process is the empty coverage document ready to be compiled.

² "Inner join creates a new set by combining the values of two sets (A and B) based upon the join-predicate." [29] In this case the two sets are the BOM and Fault Model files. The join-predicate is the Part Number of the component. The result will be a list of defect models, taken from the Fault Model file, whose part numbers are present within the BOM such that obtaining a fault modeling (subset of the *FaultModel.ini*) of the board under test.

The produced document is basically an Excel file composed of two tabs: Details Tab ([subsection 6.2.1.1](#)) and Summary Tab ([subsection 6.2.1.2](#)).

6.2.1.1 Details Tab

It reports detailed coverage information. From this document, after compilation, it is possible to obtain back the test strategies coverage results (*"Which test covers which defect?" "Is this defect uncovered?" "Is the presence of this component covered by the test line?"* and so on).

The details tab is structured as reported in [Table 15](#). Each row involves several information:

- **Reference Number:** Project-unique number reported in the schematic.
- **Value:** It is the component physical value.
- **Type:** It is the type of the component.
- **Part Number:** It is the component MM-unique number (footnote¹ at Page 81).
- **Pin Count:** Number of pins of the component.
- **Page:** Page reference of the component within the schematic. Useful for the Test Engineer during Coverage Document compilation ([subsection 6.2.2](#)).
- **Fault:** Component defect for which coverage information in the row are reported.
- **AOI:** AOI Coverage mark for the related defect (it is filled up with "X" if the defect is covered by AOI. It is left empty otherwise).
- **ICT:** ICT Coverage mark for the related defect (it is filled up with "X" if the defect is covered by AOI. It is left empty otherwise).
- **EOL:** EOL Coverage mark for the related defect (it is filled up with "X" if the defect is covered by AOI. It is left empty otherwise).
- **ALL:** Test Line Coverage mark for the related defect (it automatically green-colors itself if at least one of the test within the line covers the defect. It remains red-colored otherwise).
- **Presence:** Presence Coverage mark for the related component (it automatically green-colors itself if at least one of the test within the line covers the "Presence" defect of the component. It remains red-colored otherwise). It is needed to compute the component coverage information present in the "Summary Tab" ([subsection 6.2.1.2](#)). In fact, the number of green cells present in this column represents the number of components for which the "Presence" is detectable by the test line.
- **Notes:** Test Engineer annotations.

Ref.	Value	Type	Part Number	Pin Count	Page	Fault	AOI	ICT	EOL	ALL	Presence	Notes
R53	10k	R	NR.04.A	2	pag1.1	Presence						
R53	10k	R	NR.04.A	2	pag1.1	Alignment						
R53	10k	R	NR.04.A	2	pag1.1	Value						
R53	10k	R	NR.04.A	2	pag1.1	Short 1-2						
R53	10k	R	NR.04.A	2	pag1.1	Quality 1						
R53	10k	R	NR.04.A	2	pag1.1	Quality 2						
...						
C72	1nF	C	NC.34.O	2	pag3.4	Presence						
C72	1nF	C	NC.34.O	2	pag3.4	Alignment						
C72	1nF	C	NC.34.O	2	pag3.4	Value						
C72	1nF	C	NC.34.O	2	pag3.4	Short 1-2						
C72	1nF	C	NC.34.O	2	pag3.4	Quality 1						
C72	1nF	C	NC.34.O	2	pag3.4	Quality 2						
...						

Table 15: MM-PWT_{coverage} - Empty Document (Details Tab)

To generate the Details Tab, the Defect Universe (generated by the script) is simply copied and pasted to the Details Tab of the pre-formatted Excel file.

6.2.1.2 Summary Tab

The Summary Tab sums up the coverage info contained in the Details tab. The [Table 16](#) schematizes its structure (obviously, since it has been just generated, the coverage percentages are all set to 0%).

On the top side it is possible to notice the number of components within the board (310). The coverage in yellow here represents the capability of the test line to detect the components presence (it is basically the summary data of the "Presence" column in the Details Tab, [subsection 6.2.1.1](#)).

In the bottom table, the whole coverage is divided in meaningful schematic subsections (for example, the components belonging to the MCU input signals are resumed into a category, resistors/capacitors in another category and so on) in order to easily and quickly identify, after compilation, weaker coverage area of the board. Finally, all coverage sections are summarized into a final coverage row (blue light one) in which the Fault Coverage percentage is reported (yellow cell). It expresses the capability of the test line to detect modeled defects (in this case the whole defect

universe is composed by 2527 defects). To conclude, the "links" column allows the Test Engineer to jump to the referenced section in the Details Tab to check what components the coverage sections derive from. Differently than the Details Tab, the Summary Tab requires much more manual effort to be built.

	Components Modeled	Components Covered	Components %
COMPONENT Coverage	310	0	0%

	Faults Modeled	Faults Covered	Faults %	AOI	AOI %	ICT	ICT %	EOL	EOL %	Notes	Links
FAULT Coverage	2527	0	0%	0	0%	0	0%	0	0%		-
Communication	85	0	0%	0	0%	0	0%	0	0%		<i>Communication</i>
...
Connector	121	0	0%	0	0%	0	0%	0	0%		<i>Connector</i>
...
Input Signals	510	0	0%	0	0%	0	0%	0	0%		<i>Input Signals</i>
Logic Unit	614	0	0%	0	0%	0	0%	0	0%		<i>Logic Unit</i>
...

Table 16: MM-PWT_{coverage} - Empty Document (Summary Tab)

6.2.2 COMPILATION

Pre-conditions:

- **Empty Coverage Document:** Empty document ready to be compiled.

Post-conditions:

- **Final Coverage Document:** Document summarizing all coverage information.

Once created (Summary Tab and Details Tab completed), the document is ready to be compiled. This is the most stressful and tiresome phase which requires from 15 to 20 days to be completed since, by consulting the electrical schematic component after component, the Test Engineer compiles the document using all its own testing knowledge to reason about whether different defects could be covered by any of the various test strategies. The compilation reasoning obviously varies according to the test strategy under analysis. Next subsections describe few examples of compilation, one for each test strategy.

The pilot circuit which is taken into account is the one reported in [Figure 65](#). A brief functional analysis can be the following one: OUT is normally connected to IN1 filtered signal. Whenever IN2 activates the MOSFET contained into Q8, the OUT signal is brought to GND.

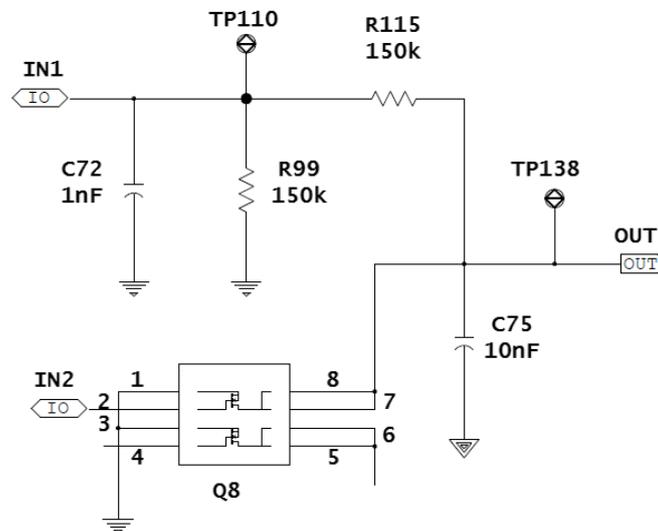


Figure 65: MM-PWT_{coverage} Document Compilation - Reference Circuit

Components under analysis are C72 and R115 but the way of operating is similar for the whole board. The fault model for any of these two elements is the following:

- **Presence (Open):** It models the physical absence in the sense that the component is not mounted at all but also the electrical absence in the sense that, if either Pin1 or Pin2 are not soldered to the board, it is electrically absent (Open Circuit).

- **Alignment:** It models the wrong alignment of the mounted component.
- **Value:** It models the wrong nominal value of the component.
- **Short 1-2:** It models the potential short between Pin1 and Pin2.
- **Quality 1:** It models the quality of the solder applied to Pin1.
- **Quality 2:** It models the quality of the solder applied to Pin2.

6.2.2.1 AOI Coverage Document Compilation

AOI can only cover for both elements the *Quality 1*, *Quality 2* and *Alignment* faults by performing a visual inspection to check solders quality and component placement.

6.2.2.2 ICT Coverage Document Compilation

The ICT compilation is based on the fact that the test is performed by contacting the TPs. The reasoning applied to compile the ICT column is the following:

- **R115:** TP110 and TP138 can be used to test R115 by applying "Resistor Test" ([subsubsection 4.2.2.4](#)). Thus, *Presence*, *Value* and *Short 1-2* are covered by ICT. *Alignment*, *Quality 1* and *Quality 2* can not be covered by the ICT electrical test.
- **C72:** TP110 and TP_GND (not shown) can be used to test C72 by applying "Capacitor Test" ([subsubsection 4.2.2.5](#)). The parallel R99//C72 is not a problem since the RC constant is big enough to get the C72 charging transient. Thus, *Presence*, *Value* and *Short 1-2* are covered by ICT. *Alignment*, *Quality 1* and *Quality 2* can not be covered by the ICT electrical test.

6.2.2.3 EOL Coverage Document Compilation

The EOL compilation is based on the fact that the test is performed by "talking" with the board through the connector. The reasoning applied to compile the EOL column is the following:

- **R115:** The OUT signal of the connector (dependent on IN1) is read by the ATE. This is a way to make the signal (transiting from IN1 to OUT) passing through R115. The *Presence* is covered in this case since a potential open circuit would imply a floating OUT. In addition, there is no way to understand if the resistor is out of tolerance (wrong value) neither if an eventual short is present between its ends because the read voltage will be the same in any case.
- **C72:** The OUT signal of the connector (dependent on IN1) is read by the ATE. This is a way to make the signal (transiting from IN1 to OUT) passing through C72 filter capacitor. In this case only the *Short 1-2* is covered since it would bring all the current to ground implying OUT to be floating. The eventual absence of C72 would surely prevent the signal to be filtered out, but this situation can not be detected by EOL.

The final compiled coverage document concerning R115 and C72 is reported in [Table 17](#). The covered defects have become green-colored to notice that the test line is capable of covering at 100% the modeled defects for these components.

Ref.	Value	Type	Part Number	Pin Count	Page	Fault	AOI	ICT	EOL	ALL	Presence	Notes
R115	150k	R	NG.099.A	2	pag1.a	Presence		X	X			
R115	150k	R	NG.099.A	2	pag1.a	Alignment	X					
R115	150k	R	NG.099.A	2	pag1.a	Value		X				
R115	150k	R	NG.099.A	2	pag1.a	Quality 1	X					
R115	150k	R	NG.099.A	2	pag1.a	Quality 2	X					
R115	150k	R	NG.099.A	2	pag1.a	Short 1-2		X				
C72	1nF	C	NC.534.O	2	pag1.a	Presence		X				
C72	1nF	C	NC.534.O	2	pag1.a	Alignment	X					
C72	1nF	C	NC.534.O	2	pag1.a	Value		X				
C72	1nF	C	NC.534.O	2	pag1.a	Quality 1	X					
C72	1nF	C	NC.534.O	2	pag1.a	Quality 2	X					
C72	1nF	C	NC.534.O	2	pag1.a	Short 1-2		X	X			

Table 17: MM-PWT_{coverage} - Compiled Document (Details Tab)

It is important to notice that all these consideration have been done for just two components of the board. Very often, boards contain huge amount of devices causing the coverage compilation to become something difficult and hard to deal with.

6.2.3 COVERAGE COMPUTATION

Finally, once the document has been compiled, the final coverage is computed. The computation is straightforward in the sense that, once the document is fully compiled, the pre-set up Excel file allow to show up the final coverage percentages in the Summary tab (as shown in [Table 16](#)), without any effort required:

- Component Coverage

It summarizes the capability of the test line to detect the presence/absence of the components within the board. MM-PWT_{component_coverage} is based on the ratio between number of components whose "Presence" fault is covered and total number of components.

$$MM_PWT_{component_coverage} \equiv \frac{\#Presence_{covered}}{\#Components} \quad (9)$$

- Fault Coverage

It summarizes the capability of the test line to detect the modeled faults (Defect Universe). $MM_PWT_{fault_coverage}$ is based on an averaged computation (Equation 5). Simply, it is obtained as the ratio between covered faults and total faults.

$$MM_PWT_{fault_coverage} \equiv \frac{\#Defects_{covered}}{\#Defects} \quad (10)$$

6.3 Advantages and Disadvantages

The $MM_PWT_{coverage}$ is characterized by having a defect universe adaptive for the board under test (thanks to the MM_PWT_{metric} properties). However, the sections "Creation" (subsection 6.2.1) and "Compilation" (subsection 6.2.2) of a Coverage Document gave an idea of how much effort, time and accuracy are needed to compile the document. These weaknesses do not permit that the described MM_PWT hardware fault coverage analysis participates in a closed validation loop during the prototype design and development phases.

6.4 Goals

It would be great having a highly automatic hardware fault coverage flow which, being fast and effortless, would help the design, the layout and the manufacturing phases:

- **Design:** An Electrical DfT checker could be performed. Design must be characterized by some properties (as for example one TP should be placed in each multiple-pins net) to allow the test.
- **Layout:** A Mechanical DfTs checker could be performed. Limits in the test equipment strictly require that some mechanical constraints are respected in order for some tests to be properly performed (as for example the minimum distance between TPs: if violated, the ICT equipment could not contact all TPs).
- **Manufacturing:** An ex-post analysis could be performed to compute the actual hardware fault coverage of the test line applied to the product (the test is performed and the final coverage is compute).

The Part iv describes the features (chapter 7), the feasibility analysis (chapter 8) and the introduced customization (chapter 9) concerning the automatic tool in charge of achieving the objectives set.

Part IV

COVERAGE AUTOMATIC TOOL

TESTWAY EXPRESS

In [chapter 6](#) the MM-PWT Test Coverage has been described highlighting benefits and drawbacks. It is from the latter that arise the necessity to migrate the current fault coverage process towards an automatic flow. In this chapter an overall picture of TestWay Express is provided ([section 7.2](#)) in order to later proceed by describing the tool-feasibility analysis performed by the company ([chapter 8](#)).

7.1 Overview

"TestWay is a proven solution, produced by Aster Technologies and used by many PCBA designers and manufacturers worldwide, that provides a unique approach to analyze electrical testability requirements and estimate test coverage aligned to specific test strategies, at the earliest opportunity in the design cycle." [20]

Traditionally, manufacturing and test constraints are only considered at the end of the layout phase, prior to production (as reported in [Figure 66](#)). This is the so called "Waterfall approach", in which Testing remains unaware about design and layout evolution. Only when these phases terminate, Testing validates the received layout. This causes the whole flow to restart again if eventual errors have to be figured out.

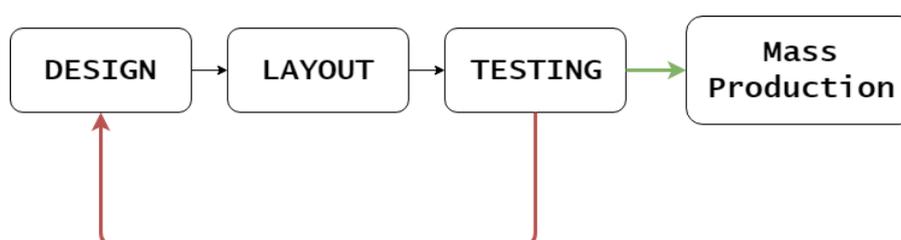


Figure 66: Production Flow - Waterfall Approach (without a Tool)

Due to board complexity, it is now mandatory to consider a validation stage at each step of the design and manufacturing phases (this kind of approach is called "V-Shape Approach", [Figure 67](#)) [21]. For this purpose, TestWay has been developed to allow users to analyze each stage from design to delivery workflow. In this way, the tool, which operates in parallel to project development, allows eventual errors in design or layout to be quickly detected and fixed (saving costs and time).

This is achieved by several features, presented at [section 7.2](#).

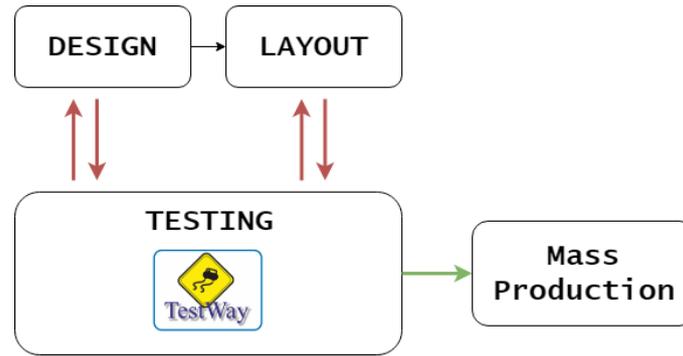


Figure 67: Production Flow - V-Shape Approach (with a Tool)

7.2 Features

In this subsections, the TestWay Express features are described one by one, particularly focusing on those aspects positively impacting on the production flow. In the Figure 68, a temporal summary of the hardware fault coverage flow is proposed.



Figure 68: TWE Features - Hardware Fault Coverage flow

7.2.1 INPUT BOARD DATA

TWE can compatibly operate with a wide variety of different CAD formats (up to 53 in the latest release) supporting layout, schematic netlist and schematic graphics. This is an added value with respect to other DfT tools, which work only from the layout stage. TWE, instead, guarantees a full interoperability between all stages across design/manufacturing flow automatically extracting from CAD files information, both geometrical and electrical, related to board design such as component

class, nominal value and tolerance range, package shape, height and access points.

The BOM file format is also supported to supplement the CAD import with some additional information such as part number, component description, value, mounted status and others.

A very powerful feature is the board schematic digitized import, which allows full interaction between reports and viewer. In other words, the imported schematic **.pdf** can become an interactive document such that any component is linked to a window showing all information related to that device.

Added Value

The possibility of importing BOM and CAD means having a continuously up-to-date version of the project under development. As mentioned in [subsection 3.1.3](#), the board is subjected to several changes during its production (components are added, removed, differently positioned and so on). After each modification, either a new BOM or a new CAD are produced. These files are used by the Testing in order to be imported into TWE so that the up-to-date version of the project can be analyzed. In [Figure 69](#) is reported the product evolution from the PROTO_A to the PROTO_B. Through the board data imports, TWE allows the Testing to be aware of the changes made during this transition.

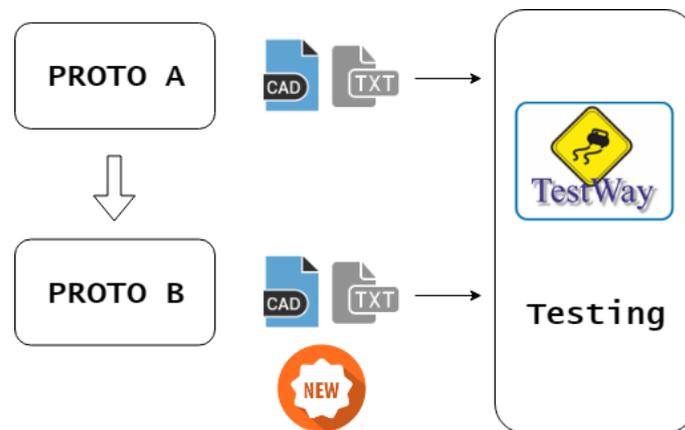


Figure 69: TWE Input Board Data - Added Value

7.2.2 MODELING THE COMPONENTS

The "component modeling" is a further step, together with the "Input Board Data" ([subsection 7.2.1](#)), to provide information to the TWE project. The main difference between them is that the files to be imported follow a strict format, thus they always provide the same properties to all projects. Whereas, concerning the component modeling, TWE makes available a user friendly editor that allows user to fill out all information not imported from BOM and CAD files. *"The better the components are modeled, the more accurate is the test strategy analysis"* [22]. It is also pos-

sible to store the component modelization into a database in order to re-use them for future projects (it is possible that same components are used in different products).

Added Value

A well-rounded board modeling allows the tool to determine the type of the test that can be conducted on the component. In this way the tool can provide quickly and automatically preventive information about various test strategies' coverage. In addition, the reusability of the modelization offered by the database storage, allows to reduce the component classification time so that coverage reports can be produced earlier (the time to be spent for those classification concerning component already classified becomes zero).

7.2.3 PLACE THE PROBES

The tests which require contact (as for example In-Circuit Test, [section 4.2](#)) presents some mechanical limits. The fixtures, above all those composed by a fixed probes configuration (bed of nails), define some constraints in terms of board accessibility. For example:

- Two probes can not be placed closer than a certain distance.
- Some areas of the board must be left free in order to be used as contrast-point when closing the fixture.
- Probes must contact a point in the board without components around.

TWE allows to define all the constraints characterizing an hypothetical tester profile in order for the user to quickly get the accessibility of that precise tester applied to that precise PCBA. The results are summarized in a text report or also visible in the Layout view.

Added Value

The added value of an automatic accessibility computation allows to:

1. Define a precise profile involving all mechanical and design for test constraints practiced by the company (DfT rules).
2. Use that profile to compute the board accessibility. It can be then analyzed and used as feedback for design and layout phase to find a compromise in the case of lack of points for the test to be accessed.
3. Apply small changes to the profile (access point types, numerical constraints and so on) to understand how the accessibility improves or worsen.
4. Perform a DfT Checker on the received Layout to detect any manufacturing constraint violation.

7.2.4 COVERAGE ANALYSIS

TWE offers the possibility of composing the production line by test strategies "drag and drop". Test strategies are either theoretical models or real testers so that both preventive and real coverage analysis can be performed, automatically and quickly, without any effort.

7.2.4.1 Coverage Estimation

The preventive coverage analysis uses a set of rules and the component classification to estimate the coverage of a particular test strategy applied to the board. This analysis is fast and automatic and gives an idea of how the test strategies can be organized to compose the best manufacturing line.

7.2.4.2 Real Coverage

Once the test has been actually developed and debugged, various test programs or coverage reports can be used to compute the real coverage adopting some standard coverage metrics such as PPVS and PCOLA/SOQ.

Added Value

The possibility of composing different manufacturing line allows to:

- Have a quick idea, during design/layout phases, of what is the detection capability of the adopted production test (*Preventive Coverage Analysis* does not require any reports/programs from real equipment and can be performed instantly as soon as BOM and CAD are available).
- Have a concrete idea of what is the quality of the adopted production test (*Real Coverage Analysis*), by importing reports/programs from real equipment.
- Produce some feedbacks to design/layout phases to remove some unneeded access points (it can be possible that the coverage analysis produces a document reporting all redundant TPs that can be removed without worsening the final coverage).
- Consult detailed coverage documents (concerning the entire test line or any single strategy) at board, component and pin levels.
- Consult graphical coverage info within the interacting Layout view.
- Produce customized report involving information desired by the company (for instance, the list of ICT uncovered pins related to MICROS or ICs constituting a starting point for developing the EOL Test Program).

TWE FEASIBILITY ANALYSIS

In the [chapter 6](#), a well-rounded description of the MM-PWT Test Coverage is reported. The laborious production of the proprietary coverage document gave birth to the need of migrating towards an automatic tool.

Given the features described at [section 7.2](#), TestWay Express was the first candidate to be adopted. The performed analysis to attest the feasibility of the tool is presented, described and its results finally analyzed.

8.1 Case of Study

The analysis is based on a comparison between two coverage documents related to the ICT test performed on a Powertrain product (SMU3 moto board):

- **TWE Real Coverage Document:** It is the coverage document resulting from a TWE Real Coverage Analysis which, as mentioned in [subsubsection 7.2.4.2](#), is performed by using *Test Programs* or *Coverage Reports* of the various techniques. In this case, the **XML ICT Test Program** has been imported into TWE to produce the real coverage document.
- **MM-PWT Coverage Document:** It is the coverage document resulting from the manual flow performed by a MM-PWT Test Engineer, as described at [section 6.2](#).

The [Figure 70](#) describes the files involved and how they are used to obtain the two coverage documents (objects of analysis).

The board, characterized by BOM and CAD files in phase of design/layout, passes through various tests during the production. In particular, the ICT test is performed by the SEICA¹ ATE running the XML Test Program. Thus, BOM and CAD files are imported into TWE to characterize the board and the XML SEICA Test Program is then used to perform the Real Coverage Analysis and finally produce the **TWE Real Coverage Document**.

Whereas, the Test Engineer manually produces the **MM-PWT Coverage Document** when it comes out from the production line.

The feasibility analysis can be now performed ([section 8.2](#)).

¹ "Founded in 1986, Seica S.p.A. is a global supplier of automatic test equipment and selective soldering systems, with an installed base of more than 1900 systems on 4 different continents." [18]

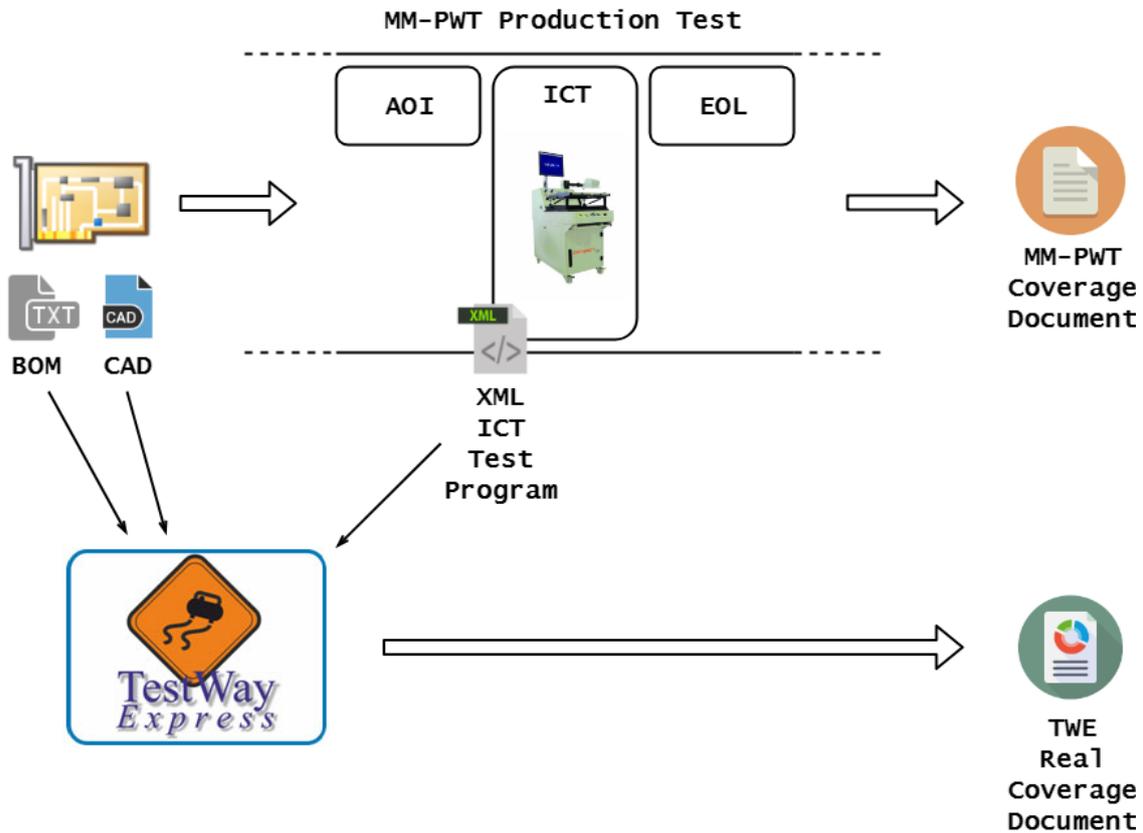


Figure 70: TWE Feasibility Analysis - Flow

8.2 Analysis

The analysis aims to answer to the following question: *"How much reliable is the TWE real coverage document?"* For this purpose, the coverage document produced by MM-PWT is used as benchmark. The process which has been followed for the analysis is now described:

1. **TWE Coverage Document Analysis:** The document produced by TWE is firstly analyzed. It contains a list of components present in the BOM. For each component, in addition to some properties, it is reported also the related PPVS coverage, result of the TWE Real Coverage Analysis based on the ICT SEICA program. This is the column which has to be analyzed.
2. **Mapping $MM-PWT_{metric}$ to PPVS:** The first problem that arises when comparing two entities is the *homogenization*. To make the documents homogeneous, the $MM-PWT_{metric}$ (described at [section 6.1](#)) has been mapped to PPVS ([subsection 5.2.2](#)).

Based on the **Rules-Compliant** property, characterizing the $MM-PWT_{metric}$ (described at [section 6.1](#)), the mapping reported in [Table 18](#) has been applied. The transformation allows to reconstruct the PPVS metric starting from $MM-PWT_{metric}$.

<i>PPVS</i>	<i>MM_PWT_{metric}</i>	<i>Mapping Rules</i>
Presence	Presence	$PPVS_{presence} = MM_PWT_{presence}$
Polarity	Polarity	$PPVS_{polarity} = MM_PWT_{polarity}$
Value	Value	$PPVS_{value} = MM_PWT_{value}$
Solder	Short, Open, Quality	$PPVS_{solder} = \frac{\sum_{covered}(MM_PWT_{open_short_quality})}{\sum_{total}(MM_PWT_{open_short_quality})}$
-	Alignment	No mapping

Table 18: TWE Feasibility Analysis - Mapping MM-PWT_{metric} to PPVS

The defined rules are described:

- **Presence:** Since presence is modeled for any component in both metrics, a straightforward mapping is performed.
 - **Polarity:** Polarity, in MM-PWT, is considered for the polarized component and ignored for the unpolarized ones. By comparing the two documents, an analogue consideration will be done for PPVS.
 - **Value:** Value, in MM-PWT, is considered for those components which can go out of tolerance in their nominal value and ignored for the other ones (for example IC, transistors and so on). By comparing the two documents, an analogue consideration will be done for PPVS.
 - **Solder:** Solder is one of the PPVS attribute which is very general. It will be compared to an average of all MM-PWT solder-considered attributes (Short between adjacent pins, Open and Quality of the solder).
 - It is worth noticing that the MM-PWT Alignment attribute, similarly to PCOLA/SOQ, does not find any association in the PPVS metric (as reported in [Table 10](#)). It will be simply not included in the comparison.
3. **MM-PWT Coverage Document Transformation:** The ICT column of the MM-PWT coverage document (structure reported at [Table 17](#)) is transformed into a PPVS-comparable one (using the rules reported at [Table 18](#)). After transformation, the resulting column is copied and pasted to the TWE real coverage document such that now the two coverages can be compared in a unique document, whose structure is reported at [Table 19](#).

ICT Coverage

REF	Presence		Polarity		Value		Solder		STATUS
	VIVA	MM	VIVA	MM	VIVA	MM	VIVA	MM	
C58	100	100	100	100	100	100	100	100	OK
D1	0	100	0	0	0	0	0	0	NO
R5	100	100	100	100	20	0	100	100	NO
DZ2	0	100	0	100	0	100	0	100	NO

Table 19: TWE Feasibility Analysis - Comparison Document

4. **Motivate the Coverage Differences:** Component after component, the comparison document is scanned trying to motivate eventual difference. The schematic is used as reference in order to technically understand which, among the two coverage evaluations, is the wrong one. The final results are summarized in [section 8.3](#).

8.3 Results

After a deep comparison has been performed, some results came out.

Most of the differences involve **files incompatibility**. Due to the board evolution process, different BOM and CAD versions exist, as well as different coverage documents. If all of them do not refer to the same board version, inconsistencies such as mounted/unmounted components or wrong nominal values can arise.

It is also important to point out that the TWE Coverage Document generation has been performed by importing BOM and CAD only, without performing any further component classification. This could have caused the tool to work with some missing information and make so some wrong assumption in terms of coverage.

However, the **results** coming from the comparison can be considered **quite satisfactory**. Nevertheless the metric heterogeneity, it has been noticed that TWE **works very well in terms of Pre-Screening Test** ([subsection 4.2.2 - "Pre-Screening Test"](#)): by using the CAD and BOM to reconstruct the electrical connections, it is capable of detecting eventual ICT-critical situation and makes related coverage assumption (filter capacitors, clamping diode parallels and some others, as reported in [subsection 4.2.3 - "ICT Limits"](#)).

TWE CUSTOMIZATION

The positive results, coming from the TWE feasibility analysis ([chapter 8](#)), made starting a process aiming to adapt the exploitability of the tool (whose features are mentioned at [section 7.2](#)) to the MM-PWT needs. Next subsections explain in details all customization applied in order to obtain a final automatic hardware fault coverage flow.

9.1 MM-PWT Customized Environment

The MM-PWT TTE team (whose responsibility and interactions are described at [section 2.2](#) and [section 2.3](#)) is going to be the final user of the tool. A suitable customized environment has been created for this purpose. It is a set of configuration files, scripts and data collected into a single folder, whose structure is resumed in [Figure 71](#) and described below.

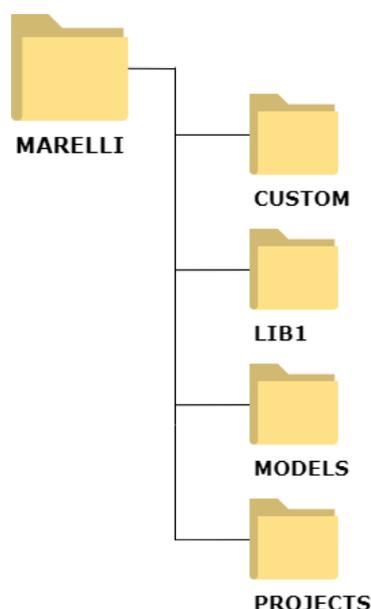


Figure 71: TWE Customization - MARELLI Folder Structure

The **MARELLI** folder contains:

- **CUSTOM**: It contains several files needed for a customized use of TWE. The content of this folder is explained step by step, in the next subsections, whenever a new customization is encountered.

- **LIB1:** It contains the files needed for a proper management of the Master Library (also called **MasterLib**). The MasterLib is a collection of component characterization grouped by Part Number¹. Details about the MasterLib are provided in [section 9.4](#), where customization related to component classification are explained.
- **MODELS:** It contains a collection of Model Files grouped by Part Number¹. A Model File resumes pin and general information of a device. Details about Model Files are provided in [section 9.4](#), where customization related to component classification are explained.
- **PROJECTS:** It contains a collection of TWE projects. Details about TWE projects and related files involved are provided in [section 9.2](#), where customization related to creation of a project are explained.

Whenever the application starts, some configuration files are loaded. The customized version of these files is contained into the CUSTOM folder. During the step "Integrate the Customized Environment" (at User Manual - [subsection A.1.1](#)) they are moved into their respective paths in order for the customized environment to be usable whenever the tool is launched. A description for any of these files is reported in next subsections.

9.1.1 BOARDVIEW.INI

It is the configuration file containing all default parameters of the viewers [23]. It is structured in sections.

In the **CONFIGURATION** section ([Listing 1](#)) two paths are set up:

- **Customer:** To let TWE refer to the customized MasterLib.
- **ModelsLibrariesPath:** To let TWE refer to the customized Models folder.

```
[CONFIGURATION]
Customer=Data\MARELLI\LIB1
ModelsLibrariesPath=VISAROOT\Data\MARELLI\MODELS
```

Listing 1: TWE Customization - BoardView.ini (CONFIGURATION)

In the **Component_Classificator** section ([Listing 9](#)), the component attributes are set. The meaning of these attributes is described at [section 9.4](#).

```

[Component_Classificator]
FieldCount=14
Field1="ERROR", "SaveNone", "Display", " "
Field2="REF", "SaveNone", "Display", "Ref"
Field3="PARTNUMBER", "SaveNew", "Edit", "Part Number"
Field4="SHAPE", "SaveNew", "Edit", "Shape"
Field5="PIN_CNT", "SaveNone", "Display", "Pin cnt"
Field6="CLASS", "SaveNew", "Edit", "Class"
Field7="VALUE", "SaveNew", "Edit", "Value"
Field8="NTOL", "SaveNew", "Edit", "NTol"
Field9="PTOL", "SaveNew", "Edit", "PTol"
Field10="SPLIT", "SaveNew", "Edit", "Split"
Field11="MODEL_FILE", "SaveNew", "Edit", "Model file"
Field12="POLARITY_MARKER", "SaveNew", "Edit", "Polarity Marker"
Field13="MOUNTED", "SaveNew", "Edit", "Mounted"
Field14="STATUS", "SaveNone", "Display", "Status"

```

Listing 2: TWE Customization - BoardView.ini (Component Classifier)

9.1.2 VISALIB.INI

It is a configuration file which allows users to define the different classes, properties and characteristics to be used for model descriptions [23]. Similarly to BoardView.ini, also this file is structured in sections.

In the **CLASSES** section the list of the classes accepted during model compilation is present:

```

CLASSES:
  DIODE      ! Diode
  EEPROM     ! EEPROM memory
  .
  .
  RESISTOR   ! Resistor

```

Listing 3: TWE Customization - visalib.ini

With the analogue syntax, **ATTRIBUTES** section involves the attributes characterizing a component during model compilation (VALUE, SHAPE and so on) whereas **PROPERTIES** section involves all attributes that can characterize a pin (IN, OUT, INOUT and so on).

Up to now, some editing to this file have been performed but it is not excluded that additional ones are needed in future (for example if new component classes are encountered).

9.2 Creation of a New Project

New project creation has been customized such that any new project is linked to precise configuration files, scripts and anything needed for the whole coverage flow to be approached as desired by MM-PWT.

For this purpose, whenever a new project is created (step "Create New Project" in the User Manual - [subsection A.1.2](#)) a script creates a directory, within

MARELLI/PROJECTS folder, with the structure shown at [Figure 72](#) and described below. It allows the project to make full use of all customization applied to the tool and described in the next sections.

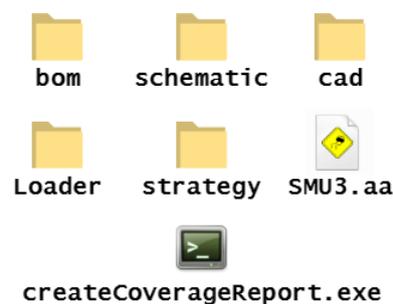


Figure 72: TWE Customization - New Project Environment Structure

- **bom**: It is the folder into which the BOM file has to be moved. It contains the files needed for the BOM to be properly imported into TWE (*BOM.grm*). Consult [subsection 9.3.1](#) for further details about BOM file import.
- **schematic**: It is the folder into which the schematic file has to be moved.
- **cad**: It is the folder into which the CAD file has to be moved. Consult [subsection 9.3.2](#) for further details about CAD file import.
- **Loader**: It is the folder into which the files for the Real Coverage Analysis have to be moved (XML program related to ICT and coverage information related to EOL - User Manual at [subsection A.1.8](#)).
- **strategy**: It contains all the files needed for the customized coverage analysis to be performed.
- **createCoverageReport.exe**: It is the script that updates the coverage report after the analysis is performed.
- **SMU3.aa**: It is the TWE project file (related in this case to the SMU3 board) which is updated by the tool each time a new configuration is applied to the project. By default, if the project is created from the TWE environment, this file is empty. Throughout a customization, a pre-formatted version of the *.aa* file is

generated, so that the just created project is characterized by all customized features, as shown in [Figure 73](#) and described below.

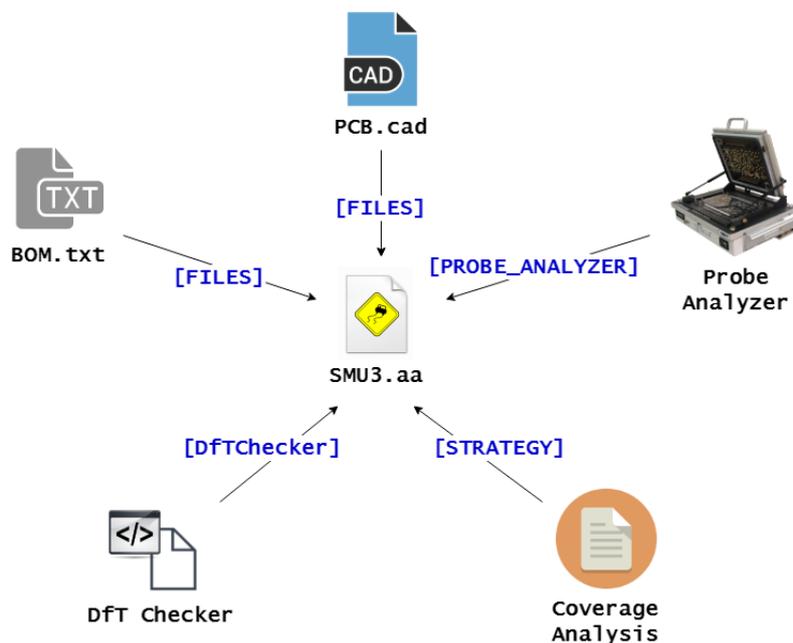


Figure 73: TWE Customization - New Project Flow-Links

The section **FILES** allows the project to be linked to BOM and CAD files contained in the respective folders (consult [subsection A.1.3](#) to have more details about customized files import).

The section **DfTChecker** allows the project to be linked to proprietary customized rules and scripts. These are used in support to other customization (such as [subsection 9.7.1](#) in support to the *Coverage Reports* generation).

The section **Probe_Analyzer** allows the project to be linked to proprietary customized profiles to compute the ICT accessibility based on MM-PWT layout rules (consult [section 9.5](#) to have more details about probe analyzer aspects).

The section **STRATEGY** allows the project to be linked to the MM-PWT Test Strategy Line and produce, according to the test strategies selected, related PCOLA/SOQ coverage reports.

9.3 Files Import

The Files Import is the step through which TWE collects all board data, which the tool uses and processes during the whole coverage flow. It is a very delicate step since some final results could be altered if not proper imports are performed.

Next subsections explain the customization introduced for both BOM and CAD files.

9.3.1 BOM FILE IMPORT

The method adopted by TWE to import data from the BOM file is based on a grammar file. Basically, information are extracted from the file and stored into a runtime database (also called Component Classifier) regarding on precise pattern matching rules (specified within the grammar file). The whole customization process is described now step by step:

1. **Company BOM-Format selection:** The FUJI² format has been selected to standardize the customization process. The reason why behind this choice is related to the fact that the BOM FUJI, being the input of the *Pick and Place* machine (*Pick and Place* - Front-End, [subsection 3.1.1](#)), will be always available to be imported very early in the production flow. The FUJI format is reported at [Table 20](#).

Ref	Label	Tol	Shape	Part Number	X	Y	Rot
C1	C_...8PF+-0,5PF_.50V	5%	C04_RW	NC.006.A	12.5	9.2	90°
R4	C_..470.KO_+-5%1/16W	5%	R04_RW	NR.031.A	-5.2	-8.9	0°
L5	C_.6,8NH..+-10%_1A	10%	L08_RW	52023616	23.5	90.4	180°

Table 20: BOM FUJI - File Format

2. **TWE-compatible BOM Generation:** The information reported in the BOM FUJI are not all needed. Some of them, such as X, Y and Rot (which are exclusively used by the Pick and Place to put the component on the board surface) are discarded. A script has been developed to process the useful information from the Fuji and move them into a new file, ready to be imported by TWE. The structure of the TWE-compatible BOM is reported at [Table 21](#).

C1	NC.006.A	8pF	-5%	+5%	C04_RW
R4	NR.031.A	470KOhm	-5%	+5%	R04_RW
L5	52023616	6.8nH	-10%	+10%	L08_RW

Table 21: BOM TWE-compatible - File Format

As it can be noticed, values are separated by the /t delimiter.

3. **Grammar File Definition:** Based on the structure of the TWE-compatible BOM, the grammar file is defined, as reported in [Listing 4](#).

```
%REF% %PARTNUMBER% %VALUE% %NTOL% %PTOL% %SHAPE%
```

Listing 4: Files Import - BOM Grammar File

In this way, for each row of the BOM, the data, separated by /t, are imported into the classifier (TWE runtime database) whenever the project is loaded, thanks to the customization reported at next step ([item 4](#)).

4. **Link BOM Import to Project:** It is important now that whenever the project is loaded, a BOM data import is performed. This is allowed by adding to the **.AA** project file the code reported at [Listing 5](#).

```
[FILES]
```

```
BOM=BOM,AA:bom\BOM.txt
```

Listing 5: TWE Customization - Link BOM Import to Project

The file **BOM.txt** is the TWE-compatible BOM, result of [item 2](#). It is important to notice that the grammar file must be present within the same path such that TWE uses the customized grammar to perform the BOM import.

The whole procedure allowing a customized BOM import is resumed in [Figure 74](#).

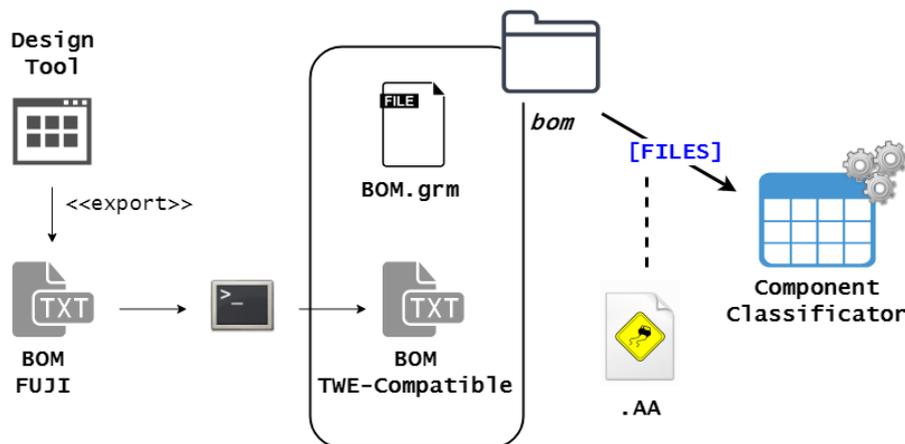


Figure 74: TWE Customization - BOM Import

9.3.2 CAD FILE IMPORT

The method adopted by TWE to import data from the CAD file is quite straightforward. The customization process is described now step by step:

1. **Select CAD Format:** Concerning the CAD File, as mentioned in [subsection 7.2.1](#), a several number of layout from native CAD formats are supported (GENCAD, CAMCAD, FATF and ODB++). The format which has been selected is the CADENCE ALLEGRO (.CAD) native one.
2. **Configure Export of CAD File:** Cadence Allegro is a CAD software which allows that requires a script to get out data of the produced layout. The templates taken in input by the script allow to specify the fields to be extracted into the final ASCII file (which is then imported into TWE). The templates adopted by MM-PWT related to component ([Listing 6](#)) and vias ([Listing 7](#)) are reported above.

```
COMPONENT  
  
REFDES  
COMP_CLASS  
COMP_PART_NUMBER  
COMP_HEIGHT  
COMP_DEVICE_LABEL  
COMP_INSERTION_CODE  
SYM_TYPE  
SYM_NAME  
SYM_MIRROR  
SYM_ROTATE  
SYM_X  
SYM_Y  
COMP_VALUE
```

Listing 6: Component Data - CADENCE ALLEGRO export

```
COMPOSITE_PAD  
CLASS="VIA CLASS"  
VIA_X  
VIA_Y  
PAD_STACK_NAME  
NET_NAME  
TEST_POINT  
VIA_MIRROR
```

Listing 7: Vias Data - CADENCE ALLEGRO export

Most of these information are useful to reconstruct the board layout. However, some others are the same of those imported by the BOM. This does not create any conflict since an import priority can be defined in the TWE setup.

3. **Link CAD Import to Project:** It is important now that whenever the project is loaded, a CAD data import is performed. This is allowed by adding to the .AA project file the code reported at [Listing 8](#).

```
[FILES]
LAYOUT=CADENCE,AA:cad\pcb.cad
```

Listing 8: TWE Customization - Link CAD Import to Project

The whole procedure allowing a customized CAD import is resumed in [Figure 75](#).

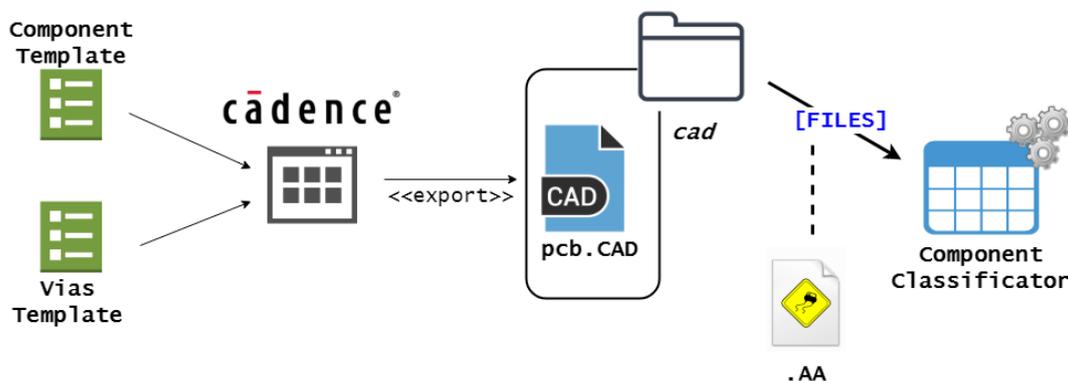


Figure 75: TWE Customization - Customized CAD Import

9.4 Component Classification

The information imported from BOM and CAD files are not enough for a well-rounded board characterization but are essential to reconstruct electrical connections between components (design) and physical geometries of the board (layout). What is missing is the full classification of any single device, which is detailed in this section.

The Component Classification is a very delicate aspect but, if properly used, allows to drastically reduce the time to generate a coverage analysis (reusability aspect mentioned at [subsection 7.2.2](#)). The [Figure 76](#) describes which are the classification levels offered by TWE.

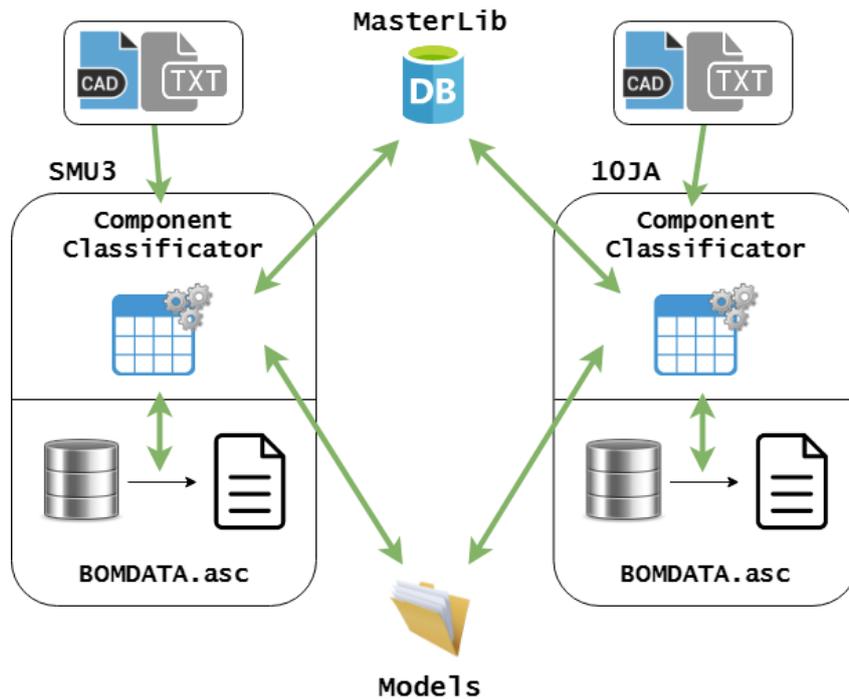


Figure 76: Component Classification - TWE Management

In the [Figure 76](#) two projects are involved (SMU3 and 10JA). Whenever the project is loaded, the BOM and CAD data as well as the ones present in Models, local and remote databases are loaded into the **Component Classifier** (the volatile runtime database). This is a user-friendly editor which allows users to edit the component classification. After the editing phase, it is possible to locally (to **BOMDATA.asc** file) or globally (to the **MasterLib** file or **MODELS** folder) store the changes such that, in future, it is possible to reuse the performed classification. Note that all unsaved changes performed within the Component Classifier are lost when closing the TWE environment.

Next subsections describe the customization introduced to properly manage and store the classification of components.

9.4.1 COMPONENT CLASSIFICATOR

The Component Classifier allows to consult and edit the properties adopted to characterize and classify the components. It is basically a grid which involves components and related properties respectively arranged in rows and columns.

The customized set of attributes, chosen to characterize the devices, is defined in the file **BoardView.ini** as reported in [Listing 9](#).

```

[Component_Classificator]
FieldCount=14
Field1="ERROR", "SaveNone", "Display", " "
Field2="REF", "SaveNone", "Display", "Ref"
Field3="PARTNUMBER", "SaveNew", "Edit", "Part Number"
Field4="SHAPE", "SaveNew", "Edit", "Shape"
Field5="PIN_CNT", "SaveNone", "Display", "Pin cnt"
Field6="CLASS", "SaveNew", "Edit", "Class"
Field7="VALUE", "SaveNew", "Edit", "Value"
Field8="NTOL", "SaveNew", "Edit", "NTol"
Field9="PTOL", "SaveNew", "Edit", "PTol"
Field10="SPLIT", "SaveNew", "Edit", "Split"
Field11="MODEL_FILE", "SaveNew", "Edit", "Model file"
Field12="POLARITY_MARKER", "SaveNew", "Edit", "Polarity Marker"
Field13="MOUNTED", "SaveNew", "Edit", "Mounted"
Field14="STATUS", "SaveNone", "Display", "Status"

```

Listing 9: TWE Customization - *BoardView.ini* (Classifier)

The list of attributes involves:

- **ERROR**: It is a purely graphical attribute, which shows whether the component is bad (red cross ) or well classified (green check ). In the former case, the column STATUS reports information about the bad classification (missing attribute, bad values and others).
- **REF**: It is the reference name of the component compliant with the electrical Schematic document.
- **PARTNUMBER**: It is the Part Number (footnote¹ at Page 81) MM-PWT of the component.
- **SHAPE**: It is a label that identifies the geometry aspects of the component.
- **PIN_CNT**: It is the component number of pins.
- **CLASS**: It is the Class¹ of the component.
- **VALUE**: It is the nominal physical value of the component.
- **NTOL**: (standing for **N**egative **T**OLerance) It is a permissible limit of negative variation in the component physical value.
- **PTOL**: (standing for **P**ositive **T**OLerance) It is a permissible limit of positive variation in the component physical value.
- **SPLIT**: It handles components made of similar sub components allowing to define their internal structure.

¹ The Class identifies a set of components united by the same electronic/electrical functionality (for instance RESISTOR, CAPACITOR, INDUCTOR, MICROPROCESSOR and so on so forth).

- **MODEL_FILE**: It is the PATH to the model file associated to the component, in which pin information are reported (consult [subsection 9.4.4](#) to have more details about Models).
- **POLARITY_MARKER**: It is a boolean value (YES or NO) which reports whether the component includes a polarity marker on its body surface or not (polarity marker figure at [Figure 21](#)).
- **MOUNTED**: It is a boolean value (YES or NO) which reports whether the component is mounted or not.
- **STATUS**: Reports information about bad component modeling.

Whenever the project is loaded, as the [Figure 76](#) shows up, all possible information are imported into the Classifier. Eventual additional changes must be saved, using the Component Classifier top bar, either in the local storage () or in the global one ().

9.4.2 LOCAL CLASSIFICATION STORAGE

The local classification storage is a file **.ASC**, automatically created by TWE when clicking on  , which keeps trace of those classification exclusively local and, thus, not stored in the MasterLib yet. No customization have been applied in this case, since TWE adopts a proper method to automatically save and retrieve local data.

9.4.3 GLOBAL CLASSIFICATION STORAGE

The global classification storage is also called **MasterLib**. It is used to store component data, from projects, to be used on future projects within the Component Classifier. The MasterLib content follows a strict syntax, specified into a grammar file (concept similar to the BOM file import - [subsection 9.3.1](#)). The involved customization process is described now step by step:

1. **Attributes Definition**: The MasterLib is a container of device classification. The attributes to be stored must be exclusively related to the components and not to the correlation that the component has got with a precise project (such as accessibility or coverage information). The attributes chosen are a subset of those involved in the Component Classifier (reported at [9.4.1](#)): **PART-NUMBER** to identify uniquely the component, **CLASS** to identify its category, **VALUE**, **PTOL** and **NTOL** to specify physical aspects, **SHAPE** and **POLARITY_MARKER** to reports geometrical characteristics and finally **PIN_CNT**, **MODEL_FILE** and **SPLIT** to describe both internal and external structures of the component.

2. **Grammar File Definition:** The file **MASTERLIB.GRM** involves the code reported at [Listing 10](#).

```
PARTNUMBER;CLASS;MODEL_FILE;SHAPE;PIN_CNT;VALUE;PTOL;NTOL;SPLIT;  
POLARITY_MARKER  
%key_partnumber%;%class%;%model_file%;%shape%;%pin_cnt%;%value%;%ptol%;%ntol%  
;%split%;%polarity_marker%
```

Listing 10: TWE Customization - MasterLib Grammar File

3. **MasterLib Header Definition:** Before starting populating the MasterLib file, it is important to define the MasterLib compliant with its grammar. Thus, the file **MASTERLIB.CSV** is created with the header reported at [Listing 11](#).

```
PARTNUMBER;CLASS;MODEL_FILE;SHAPE;PIN_CNT;VALUE;PTOL;NTOL;SPLIT;  
POLARITY_MARKER
```

Listing 11: TWE Customization - Header MasterLib File

4. **Link MASTERLIB to the Project:** It is important now that whenever the project is loaded, a MasterLib data import is performed and whenever the user desires to globally save the performed classification, a MasterLib data export is performed. This is allowed by adding to the **BoardView.ini** file the code reported at [Listing 12](#).

```
[CONFIGURATION]  
Customer=Data\MARELLI\LIB1
```

Listing 12: TWE Customization - Link MASTERLIB to the Project

The whole procedure allowing a customized MASTERLIB management is resumed in [Figure 77](#).

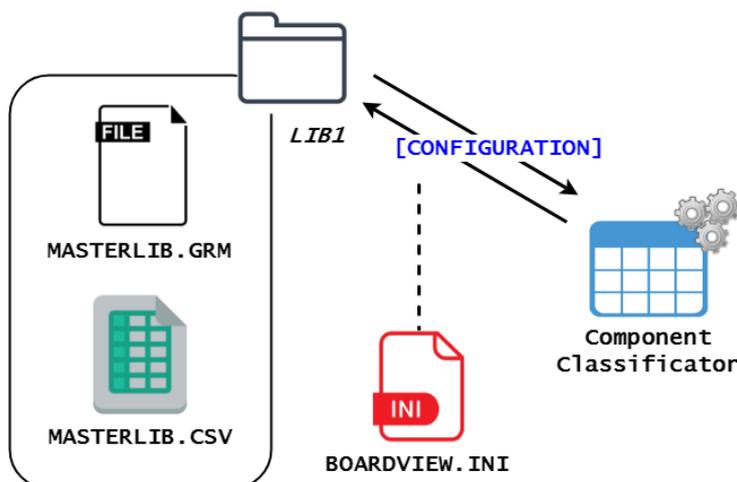


Figure 77: TWE Customization - Customized MASTERLIB Management

9.4.4 MODELS

Another global storage, made available by TestWay Express, is the Models management. Differently than the MasterLib, Models are arranged in different files, one per Part Number, in which pin level information are specified. Thus, for any Part Number there are at most one dedicated line within the MasterLib file and one dedicated file within the Models folder. The customization process allowing a customized Models management is described now step by step:

1. **Model Format Selection:** TWDES, the format offered by TWE, has been selected. The reason behind this choice is related to the fact that the tool allows to manage TWDES models throughout a user-friendly interface.
2. **Link Models Library to the Project:** In order for the Models to be properly imported into the Component Classifier, the **BoardView.ini** file has been modified by adding the code reported at [Listing 13](#).

[CONFIGURATION]

```
ModelsLibrariesPath=VISAROOT\Data\MARELLI\MODELS
```

Listing 13: TWE Customization - Link Models to the Project

In this way, whenever the Component Wizard is launched in order to look for model-components matches (as described in the User Manual - [Figure 103](#)), TWE search the models into the specified path folder.

3. **Local and Global Models:** As previously mentioned, Models allow to specify pin information: type of the pin (IN, OUT, INOUT, VDD, GND or NC²) and additional properties (whether it is a CLK signal pin, a CS and so on so forth). Note that some of these properties do not hold for all instances of the component (it can be that a pin is considered NC for a project and IN for another). Thus, it is strongly suggested to store only the full characterized models (with all pins connected) into the global directory (**Models**) and leave the ones with specific pin configuration into the local folder (**TWDES**), where TWE automatically stores them. [Figure 78](#) clarifies this concept.

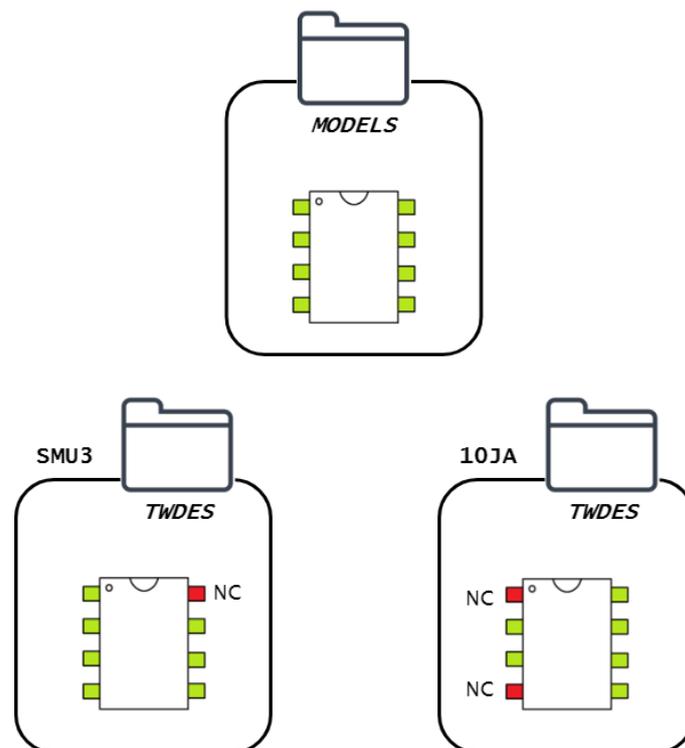


Figure 78: TWE Customization - Local and Global Models

To conclude, note that the **Models** folder is only used by TWE to look for existing models. In fact, whenever a new model is created, it is automatically stored into **TWDES** sub-folder of the active twExpress project path. Unfortunately, this is a default setting and can not be customized. Thus, eventual global models have to be manually copied and pasted from **TWDES** to **Models** folder.

4. **User Support Rule:** During classification phase, the user must have an idea of what components need a model (sometimes boards involve thousand of devices

² Not Connected Pin: It is a Pin which is voluntarily not soldered to the PCB. It is not possible, in any way, to drive or read the signal it carries.

and it is not easy to address this problem). A suitable rule has been implemented for this purpose. Whenever invoked, it returns a list of those mounted ICs for which a model has still to be created. The code of the Rule is reported at [Listing 14](#) and contained into the `MM_PWT_DfT_custom.rul` file.

```
Rule Missing_models:

Device (IC) criteria (model_file == "" && not_mounted == 0)
is reported.
```

Listing 14: TWE Customization - Missing Models Rule

In order for this file to be produced whenever a DfT Checker Analysis is performed, it is necessary to link the rule to the .AA TWE project, as reported in [Listing 15](#).

```
[DfTChecker_Rules]
Count=3
RULE0=1,VISAROOT\Data\MARELLI\CUSTOM\MM_PWT_DfT_custom.rul
! . . .
```

Listing 15: TWE Customization - Link Test Line Script to Project

9.5 Probe Analyzer

The *Probe Analyzer* allows user to place probes according to a wide variety of options [23]. It is a TWE tool that offers the possibility to define an ICT tester profile and analyze its accessibility on the real board, as reported in [Figure 79](#).

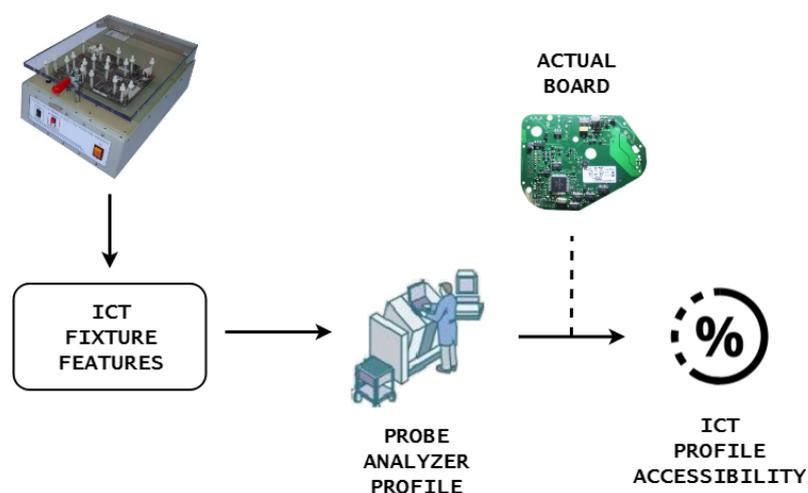


Figure 79: Probe Analyzer - Concept

The idea is to customize the Probe Analyzer such that having a tool to validate the Design ([subsection 9.5.2](#)) and Layout ([subsection 9.5.1](#)) phases, according to the Testing requirements.

Both Layout and Design for Test validation are obtained throughout a customization of the Probe Analyzer based on an official Magneti Marelli document, titled "*PRODUCT DESIGN for MANUFACTURING GUIDELINE*", whose scope is to *provide standard rules, valid for the development of ICT in Powertrain*.

9.5.1 LAYOUT FOR TEST VALIDATION

The idea is to create a customized ICT profile which exactly reflects the MM-PWT PCBA manufacturing rules in order to check whether these rules are respected in the layout of the board (if a layout rule is violated, it would appear as a non-accessibility in the Probe Analyzer result), as reported in [Figure 80](#).

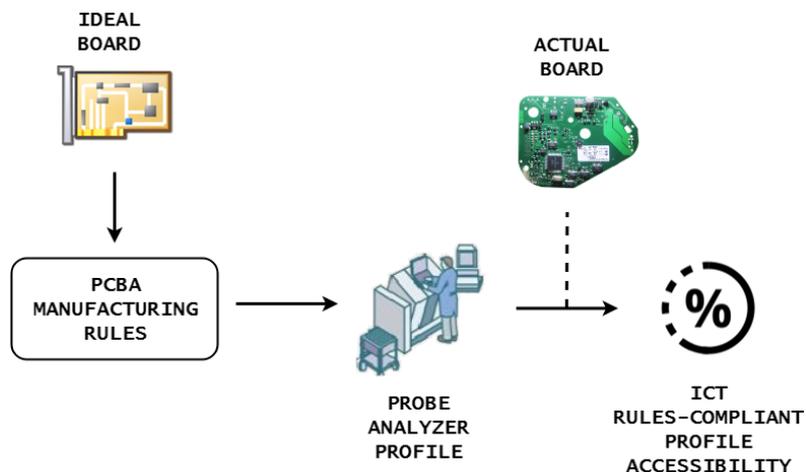


Figure 80: Probe Analyzer - Concept

The customization of the Probe Analyzer, allowing a Layout for Test validation, is composed by the following phases:

1. **Manufacturing Rules Definition:** It is first a study of the MM-PWT manufacturing rules. As result, a list of layout constraints is collected (phase described at [subsubsection 9.5.1.1](#)).
2. **Probe Analyzer Profile Creation:** It consists in adapting the collected rules into a single Probe Analyzer profile in order to be used for Layout validation (phase described at [subsubsection 9.5.1.2](#)).

9.5.1.1 Manufacturing Rules Definition

This section resumes and describes the set of rules adopted by MM-PWT in order for a produced PCBA to be properly tested by an ICT equipment.

- **Profiles of Access:** It is recommended to place all the Test Points on one side of the PCB in order to assure a proper placement of contrasts, to reduce the risk of strain and to reduce fixture weight (BOTTOM side). However, for exceptional cases the Test Points can be distributed on both surfaces (DUAL sides).
- **Minimum TP Diameter:** Regarding on the PCB components density, different constraints are set concerning the minimum diameter of the TPs, as reported in [Table 22](#).

PCB Components Density	Minimum TP Diameter
Standard	2.21 mm
High Density	1.95 mm
Very High Density	1.12 mm

Table 22: MM-PWT Manufacturing Rules - Minimum TP Diameter

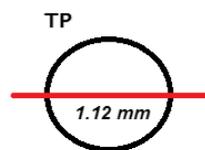


Figure 81: MM-PWT Manufacturing Rules - Minimum TP Diameter

- **Minimum Distance between TPs:** Regarding on the Probe Types, different constraints are set concerning the minimum distance between TPs, as reported in [Table 23](#).

Probe Type	Minimum Distance between TPs
GKS 100	3.0 mm
GKS 075	2.5 mm

Table 23: MM-PWT Manufacturing Rules - Minimum TPs Distance

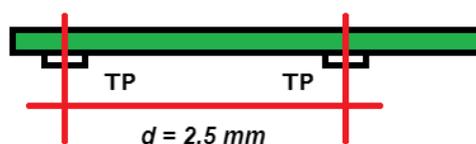


Figure 82: MM-PWT Manufacturing Rules - Minimum TPs Distance

- **Component Body Clearance:** Regarding on the component height, different constraints are set concerning the minimum clearance area (keepout border area around the component body), as reported in [Table 24](#).

Component	Minimum Clearance Area
Height < 4 mm	1.5 mm
Height > 4 mm	4.0 mm
Height > 20 mm	6.0 mm
Motor Shaft	7.0 mm

Table 24: MM-PWT Manufacturing Rules - Component Body Clearance

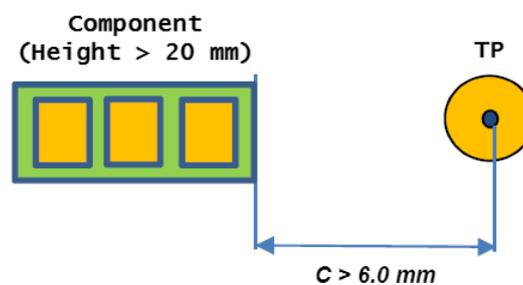


Figure 83: MM-PWT Manufacturing Rules - Component Body Clearance

- **TP/Tooling Hole Clearance:** The product design, in the placement of Test Points, must consider a clearance of 3.0 mm all around the tooling reference holes in order to avoid mechanical interference (the distance is measured from the center of the TP to the hole boundary).

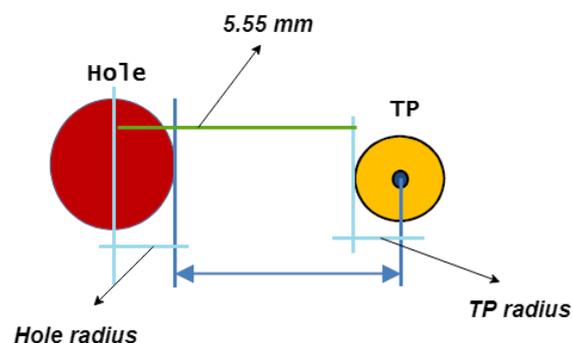


Figure 84: MM-PWT Manufacturing Rules - Tooling Hole Clearance

- **Board Outline Clearance:** The clearance measured between PCB edge and the TP center must be greater than 6.5 mm.

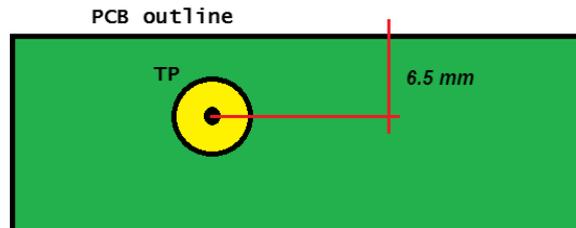


Figure 85: MM-PWT Manufacturing Rules - Board Outline Clearance

9.5.1.2 Probe Analyzer Profile Creation

In this section the customized MM-PWT ICT profile, compliant with the MM-PWT manufacturing rules (defined at [subsubsection 9.5.1.1](#)), is created in order to be used to compute the accessibility of the board resulting from the Layout validation.

The Probe Analyzer customization concerning the Layout Checker profile are now described one by one:

- **Profile:** From this Tab, it is possible to choose a pre-formatted Tester profile to compute the board accessibility. The idea is to create a customized Tester containing all the MM-PWT manufacturing rules so that can be used for the Probe Analysis.

For this purpose, the file:

MM PWT - ICT - TPs Layout Checker.pa

has been created within the folder **strategy** contained into the TWE installation root. The content of this file is divided in sections, each one referring to a profile (BOTTOM, TOP or DUAL).

Accordingly to the "**Profiles of Access**" constraint, described at [subsubsection 9.5.1.1](#), two profiles have been defined.

```
[BOTTOM]
Description=Marelli PWT Layout Checker\nBOTTOM layer only
! Profile ICT BOTTOM, Customized Constraints here

[DUAL]
Description=Marelli PWT Layout Checker\nBOTTOM and TOP layers
! Profile ICT DUAL, Customized Constraints here
```

Listing 16: TWE Customization - Layout Checker Profiles (PA)

After that customization, it would be possible to select the customized profile from the Probe Analyzer main windows, as reported in [Figure 86](#).

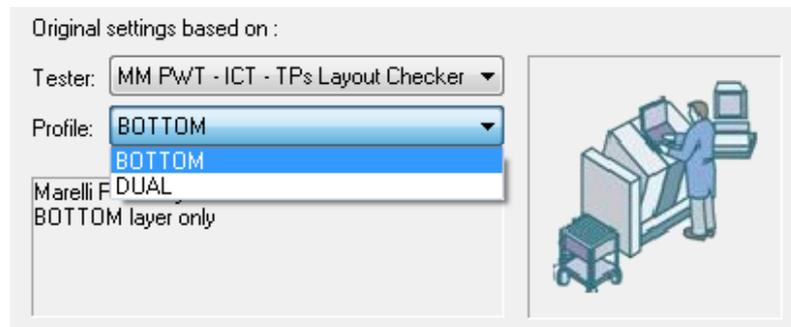


Figure 86: Probe Analyzer – Customized Profile (Layout Checker)

- **Priority Rules:** The priority rules allow to specify which parts of the board can be contacted by the customized Tester during ICT.

For this purpose, Test Points only at both BOTTOM and DUAL profiles have been specified. The customization is reported at [Listing 17](#).

```
[BOTTOM]
RulesCount=1
Rule0=Bottom;Test_Point;Default

[DUAL]
RulesCount=2
Rule0=Bottom;Test_Point;Default
Rule1=Top;Test_Point;Default
```

Listing 17: TWE Customization – Layout Checker Priority Rules (PA)

- **Constraints:** All mechanical constraints of the Tester profile are specified in this point. These customization hold for both BOTTOM and DUAL profiles because the MM-PWT rules do not make any mechanical distinction between bottom and top surfaces.
 - a. **Tester Type:** It requests whether the tester is *Fixtureless* (flying probes) or with *Fixture* (bed of nails). In this case, the adopted ICT is Bed of Nails.
 - b. **Minimum Feature Size:** *It is the minimum Access Point size below which the Probe Analyzer considers that it is not possible to place a probe [23].* Since the unique access point defined by MM-PWT is the Test Point, this parameter is set at the minimum Test Point diameter allowed.

- c. **Package Outline Clearance:** *It is a keepout area around the Nail clearance where the Probe Analyzer considers that it is not possible to place a probe or a nail [23].* This parameter is automatically computed by TWE without need of any further customization.
- d. **Board Outline Clearance:** *It is a keepout area around the circuit board [23].* This parameter is set as specified in the **Board Outline Clearance** (MM-PWT Manufacturing Rule, at [subsection 9.5.1.1](#)).
- e. **Feature Clearance:** *This defines the minimum distance between the center of two nails or probes [23].* This parameter is set as specified in the **Minimum Distance between TPs** (MM-PWT Manufacturing Rule, at [subsection 9.5.1.1](#)).
- f. **Tooling Hole Clearance:** *This defines a keepout area around the holes where no components can be placed [23].* This parameter is set as specified in the **TP/Tooling Hole Clearance** (MM-PWT Manufacturing Rule, at [subsection 9.5.1.1](#)).

The above-described customization are resumed in the [Listing 18](#). For any of them, a comment is used to associate the customization to the item of the list.

```

TesterType=Fixture           !(a.)
MinimumFeatureSize=1.12     !(b.)
FeatureClearance=2.5        !(e.)
ToolingHoleClearance=5.55   !(f.)
BoardOutlineClearance=6.5   !(d.)

```

Listing 18: TWE Customization - Manufacturing Constraints

9.5.2 DESIGN FOR TEST VALIDATION

The idea is to create a customized ICT profile which exactly reflects a MM-PWT PCBA design rule in order to check whether this rule is respected in the layout of the board (if the design rule is violated, an inaccessible net would appear in the Probe Analyzer result), as reported in [Figure 87](#).

It is crucial noticing that the layout is used to perform a design check. Theoretically, it is better using a *netlist* or a digitized version of the *electrical schematic* to perform a design validation, but such licenses have not been acquired by MM-PWT.

Drawback: It is necessary to wait for the end of the Layout phase before performing a design validation.

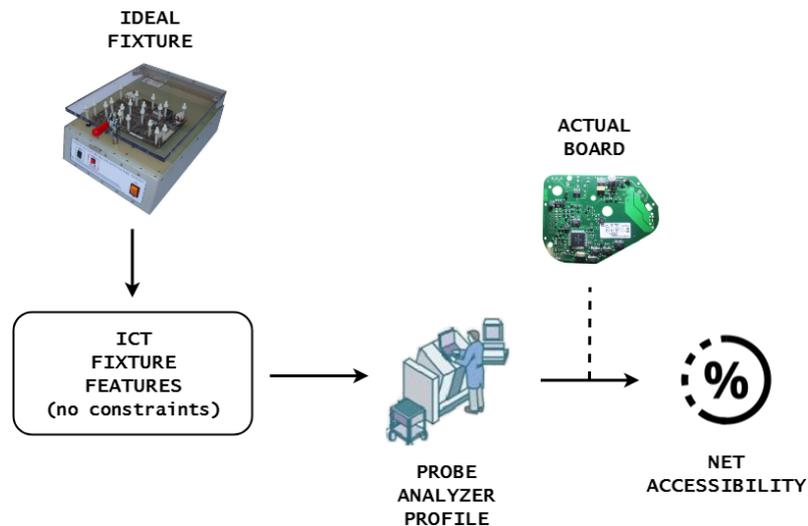


Figure 87: Probe Analyzer - Design Validation Concept

The customization of the Probe Analyzer, allowing a Design for Test validation, is composed by the following phases:

1. **Design Rule Definition:** It is first a study of the MM-PWT design for manufacturing rules document in which the design constraint is retrieved (phase described at [subsubsection 9.5.2.1](#)).
2. **Probe Analyzer Profile Creation:** It consists in adapting the design rule into a single Probe Analyzer profile in order to be used for Design validation (phase described at [subsubsection 9.5.2.2](#)).

9.5.2.1 Design Rule Definition

This section resumes and describes the rule adopted by MM-PWT in order for a designed PCBA to be properly tested by an ICT equipment.

- **Profiles of Access:** It is recommended to place all the Test Points on one side of the PCB in order to assure a proper placement of contrasts, to reduce the risk of strain and to reduce fixture weight (BOTTOM side). However, for exceptional cases the Test Points can be distributed on both surfaces (DUAL sides).
- **Access Points:** *The single Test Point must be granted for each net (100%) present in the Electrical Schematics of the Product. The presence of 2 TPs on the same net is instead necessary to realize a Kelvin Test ([subsubsection 4.2.2.4 - Low Value Resistor](#)).* It means that, except for some particular cases, in which design and manufacturing reach an agreement, each net must be accessible from at least one Test Point, sometimes two. Moreover, in case of Functional Safety Product, no TP reduction is allowed. No other constraints are defined for the ICT points of contact.

9.5.2.2 Probe Analyzer Profile Creation

In this section the customized MM-PWT ICT profile, compliant with the MM-PWT design rule (defined at [subsection 9.5.2.1](#)), is created in order to be used to compute the accessibility of the board resulting from the Design validation.

The Probe Analyzer customization concerning the Design Checker profile are now described one by one:

- **Profile:** From this Tab, it is possible to choose a pre-formatted Tester profile to compute the board accessibility. The idea is to create a customized Tester containing the MM-PWT design rule so that can be used for the Probe Analysis.

For this purpose, the file:

MM PWT - ICT - Net Accessibility.pa

has been created within the folder **strategy** contained into the TWE installation root. The content of this file is divided in sections, each one referring to a profile (BOTTOM, TOP or DUAL).

Accordingly to the "Profiles of Access" constraint, described at [subsection 9.5.2.1](#), two profiles have been defined.

```
[BOTTOM]
Description=Marelli PWT Net Accessibility Computation\nBOTTOM layer only
! Profile ICT BOTTOM, Customized Constraints here

[DUAL]
Description=Marelli PWT Net Accessibility Computation\nBOTTOM and TOP layers
! Profile ICT DUAL, Customized Constraints here
```

Listing 19: TWE Customization - Design Checker Profiles (PA)

After that customization, it would be possible to select the customized profile from the Probe Analyzer main windows, as reported in [Figure 88](#).

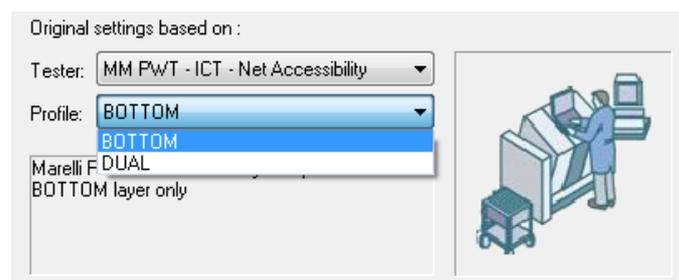


Figure 88: Probe Analyzer - Customized Profile (Design Checker)

- **Priority Rules:** The priority rules allow to specify which parts of the board can be contacted by the customized Tester during ICT.

For this purpose, Test Points only at both BOTTOM and DUAL profiles have been specified. The customization is reported at [Listing 17](#).

```
[BOTTOM]
RulesCount=1
Rule0=Bottom;Test_Point;Default

[DUAL]
RulesCount=2
Rule0=Bottom;Test_Point;Default
Rule1=Top;Test_Point;Default
```

Listing 20: TWE Customization - Design Checker Priority Rules (PA)

- **Constraints:** The only rule concerning the design involves the obligation to have at least one Test Point per net (already specified in the previous point "**Priority Rules**"). Mechanical constraints do not concern the Design phase. Thus, the customization of the Probe Analyzer concerning Design Validation is reported at [Listing 21](#).

```
TesterType=Fixture
MinimumFeatureSize=0.0
FeatureClearance=0.0
ToolingHoleClearance=0.0
BoardOutlineClearance=0.0
```

Listing 21: TWE Customization - Design Constraints

By considering all mechanical constraints at *zero*, the resulting ICT profile is ideal in the sense that it can access to any Access Point (in this case just Test Points as specified in [Listing 20](#)). Any inaccessible net, listed after the Probe Analyzer run with such profile, would not be related to mechanical features (and so to layout violation) but to an absence of Test Points (design rule violation).

9.6 Test Coverage

Test Coverage is the TWE process through which preventive and real coverage analysis are performed on the adopted test line according to all data, characterizing the project board, provided to the tool in the previous phases of the flow (such as *Files Import*, *Probe Analysis* and so on).

	AOI	ICT	EOL
<i>Preventive</i>	AOI AXI	ICT	
<i>Real</i>		VIVA SEICA SPEA3030	
<i>Declaration Method</i>			FCT

Table 25: TWE Customization - MM-PWT Test Line Coverage Analysis

Table 25 provides an overview of the possible coverage analysis typologies that MM-PWT can perform, in relation to their adopted Test Line (chapter 4):

- **AOI:** The Automated Optical/X-ray Inspection test coverage analysis is available only in *Preventive* mode (a set of customized rules are defined for both AOI and AXI techniques). No *Real* analysis can be performed since the imports for the automated inspection equipment has not been licensed from MM-PWT. No *Declaration Method* (footnote³ at Page 164) analysis can be produced since AOI and AXI tests are purely automatic image processing tests and so it is not role of the Test Engineer to define the AOI coverage.
- **ICT:** The In-Circuit test is both available in *Preventive* mode (a set of general rules are defined) and *Real* mode (for which two imports for two ICT equipment, used in production phase, have been licensed: VIVA SEICA and SPEA3030). The latter one allows to import ICT program files coming from the existing machinery involved in order to compute a real coverage.
- **EOL:** The nature of the Functional Test (adopted at End of Line by MM-PWT), based on the human reasoning, makes not possible the definition of a set of rules (*Preventive*), much less the interpretation of an ATE program or report (*Real*), for the coverage computation. Thus, TWE offers the possibility to import a set of files in which the Test Engineer himself *declares* the coverage values for all component involved in the test sequence (*Declaration Method*).

In the next subsections, the process performed by TWE and the customization introduced concerning *Preventive* (subsection 9.6.1), *Real* (subsection 9.6.2) and *Declaration Method* (subsection 9.6.3) Coverage Analysis are described.

9.6.1 PREVENTIVE COVERAGE ANALYSIS

Preventive Coverage Analysis consists in providing a test strategy coverage prediction based on a theoretical test equipment. The flow adopted by TWE to perform such estimation is reported at Figure 89 and described below.

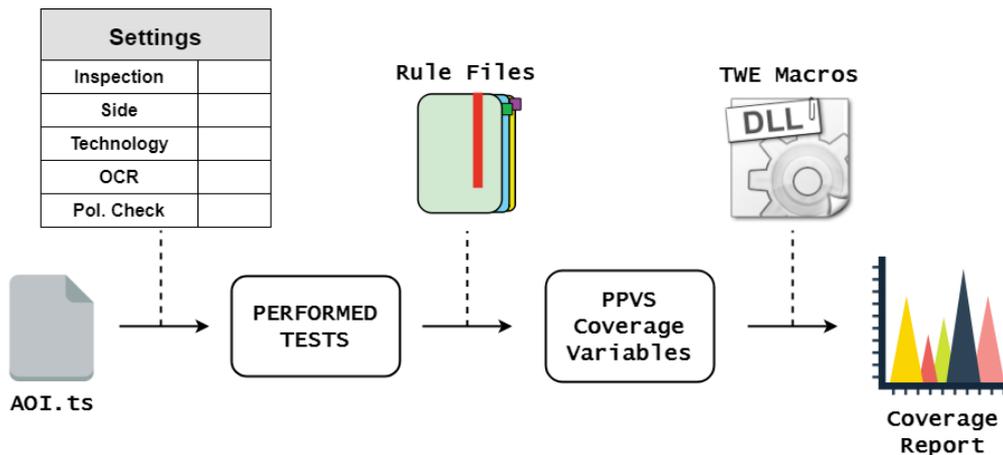


Figure 89: TWE - Preventive Coverage Analysis Flow

Whenever a preventive coverage process is launched, TWE invokes the **.TS** file related to the test under analysis. Based on the **configuration settings**, set by the user in order to obtain the desired test equipment, the **performed tests** are identified. At this point, the **test strategy rules** are applied to define which is the coverage impact of these tests on the project board. In particular, natively, the **PPVS coverages** at component and pin levels are produced. Finally, the TWE Macros process the PPVS values in order to produce the **coverage reports**.

The limits in this flow are related to the **TWE Macros** which take the PPVS Coverages in order to produce the coverage report. There are not Macros, in the previous current release, that work in PCOLA/SOQ and, unfortunately, the provided ones are not customizable. For this reason, another way has been taken into consideration, as reported in [Figure 90](#).

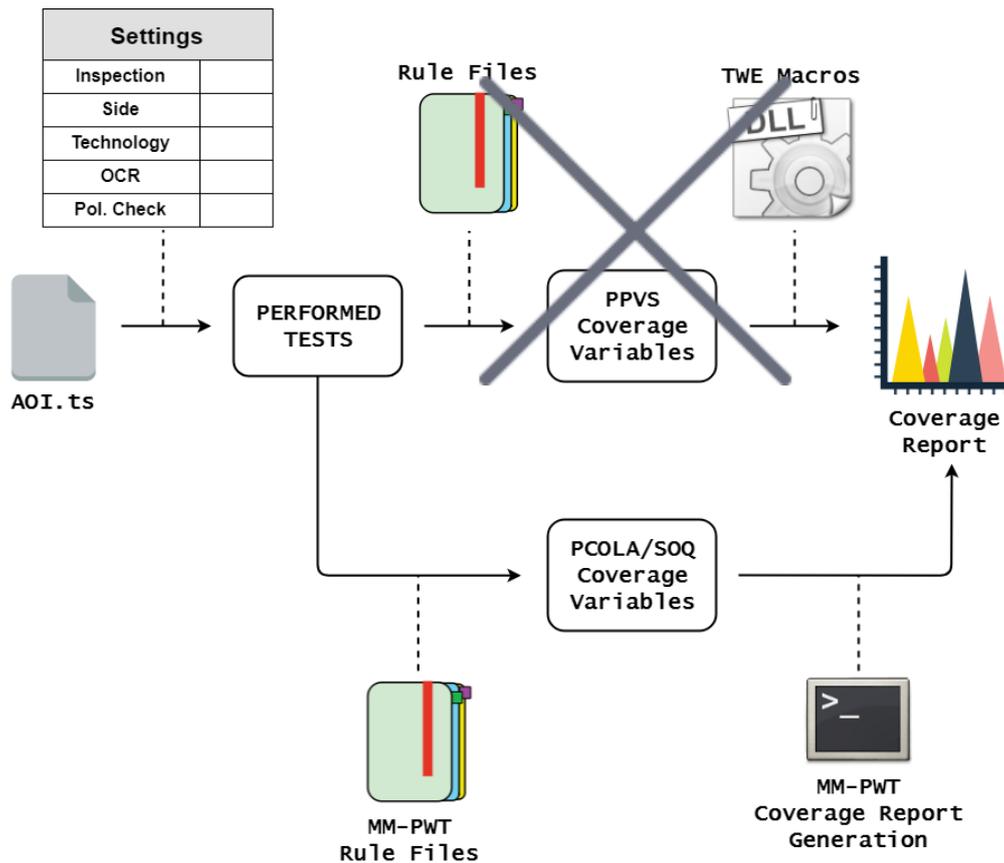


Figure 90: TWE - Customized Preventive Coverage Analysis Flow

In practice, the Rule Files have been customized in order to produce PCOLA/SOQ coverage values, in turn manipulated by a proprietary script in order to produce the final reports.

The whole customized flow, focusing on the ICT coverage estimation for a normal RESISTOR (1k), is described step by step:

1. **Test Equipment Characterization:** The default settings characterizing the preventive ICT equipment involves the customization at [Listing 22](#).

```

set test_setup("ICT", "Access", "{Full,Test_Point}") = "Test_Point"
set test_setup("ICT", "Unpowered_Analog", "{Yes,No}") = "Yes"
set test_setup("ICT", "Powered_Analog", "{Yes,No}") = "Yes"
set test_setup("ICT", "Digital", "{Yes,No}") = "Yes"
set test_setup("ICT", "Diode_Scan", "{Yes,No}") = "Yes"

```

Listing 22: TWE Customization - Preventive ICT Default Settings

As far as it can be noticed, the modeled ICT equipment is capable of performing **Unpowered Analog**, **Powered Analog**, **Digital** and **Diode Scan** macro tests accessing the board through Test Points.

2. **Performed Tests Definition:** In this phase, a rule is used to determine whether the **Resistor Test**, implemented by the **Unpowered Analog** feature, can be performed or not. In the positive case, regarding on the RESISTOR characteristics, some additional coverage information are set, as reported in [Listing 23](#).

```

if (test_setup("ICT", "Unpowered_Analog") == "Yes")

    !+
    ! Measure resistor smaller than 10 MOhms
    !-
    Device criteria (function == "RESISTOR" && value <= 10e+06) ->
        Test Resistor

    !+
    ! Can't check resistor value between 0 and 1 Ohm
    !-
    Device criteria (function == "RESISTOR" && value > 0 && value <= 1) ->
        Test -= Value

end

```

Listing 23: TWE Customization - Resistor Test Preventive ICT

The **Resistor Test** can be performed if and only if the Resistor has a value smaller than 10 MOhms. In addition, in the case of value between 0 Ohm and 1 Ohm, the equipment is considered not to be capable of detecting the resistor value (as mentioned in [subsection 4.2.3](#)). Fortunately, these parameters are configurable. There is thus the possibility of modeling an ICT equipment that best represents the reality.

The RESISTOR under analysis has a value of 1kOhm. When, during the .TS file execution, the instruction Test Resistor is encountered, the string "Resistor" is aggregated to the ICT variable associated to that component in order to notify, to the following phases, that the **Resistor Test** is performed on it. At the end of this phase, each component has associated its own ICT variable resuming, in form of strings, the ICT tests performed on it.

3. **PCOLA/SOQ Coverage Assignment:** At this point, the rule files are invoked to define the PCOLA/SOC Coverage values, based on the tests performed on each component (defined in the previous phase through the .TS file). The customization applied to the Rule files are strongly related to the knowledge consolidated in terms of ICT techniques ([subsection 4.2.2](#)). The coverage rule related to the *OPEN* and *SHORT* is defined at [Listing 24](#): Short and Open are covered for those accessible pins of all RESISTORs which have the "Resistor" string contained into their ICT variables (<< is the operator "contained into"). The criteria `level("ict") == 16` represents the accessibility pin feature (if the pin is accessible it is characterized by level 16, the highest one).

```

!+
! Resistor test
!-
Assign criteria (coverage_open_ict = 100.0) to pin when:
  Pin criteria (level("ict") == 16)
    of device (RESISTOR) criteria (ICT << "Resistor")
      is dependent on pin
        criteria (level("ict") == 16).

Assign criteria (coverage_short_ict = 100.0) to pin when:
  Pin criteria (level("ict") == 16)
    of device (RESISTOR) criteria (ICT << "Resistor")
      is dependent on pin
        criteria (level("ict") == 16).

```

Listing 24: TWE Customization - *Open* and *Short* Rules Preventive ICT

The coverage rule related to *PRESENCE* is defined at Listing 25: the Presence is detected if at least one solder is covered (`coverage_solder("ict") > 0.0`).

```

!+
! Presence detected if at least one solder is covered
!-
Assign criteria (coverage_presence_ict = 100.0) to device when:
  Device criteria (coverage_solder("ict") > 0.0)
    is reported.

```

Listing 25: TWE Customization - *Presence* Rule Preventive ICT

The coverage rule related to *CORRECT* is defined at Listing 26: the Correct is detected if and only if the Presence is detected and at the same time the Resistor Test can be performed (in case of very small or very big value the test is not effective and there is no way to understand whether the resistor is the expected one or not).

```

!+
! RLC
!-
Assign criteria (coverage_correct_ict = 100.0) to device when:
  Device (RESISTOR)
    criteria (coverage_presence_ict == 100.0 && ICT << "Resistor")
      is reported.

```

Listing 26: TWE Customization - *Correct* Rule Preventive ICT

The coverage rule related to *LIVE* is defined at [Listing 27](#): the Live is detected if and only if the Presence is detected. It is enough to attest that resistor is present in order to decree whether it is able to work (no matter how).

```
Assign criteria (coverage_live_ict = 100.0) to device when:  
Device (RESISTOR)  
  criteria (coverage_presence_ict == 100.0)  
  is reported.
```

Listing 27: TWE Customization - *Live* Rule Preventive ICT

Finally, *ORIENTATION* is ignored for resistors (since even if reversed, they still properly work) whereas *ALIGNMENT* and *QUALITY* are attributes that can not be covered by ICT test.

4. **Coverage Report Generation:** At this point, the PCOLA/SOQ component and pin variables are assigned. The last step provides that these values are processed by a script such that producing the final reports. The customization involving this phase is reported in the [section 9.7](#).

9.6.2 REAL COVERAGE ANALYSIS

Real Coverage Analysis present some limits in the current TestWay Express Release for what concerns PCOLA/SOQ metric. The problem is detailed in [subsection 10.1.2](#).

9.6.3 DECLARATION METHOD COVERAGE ANALYSIS

Declaration Method Coverage Analysis allow Test Engineers to declare coverage information related to each step of the test sequence. These information, contained into three files, are imported by TWE through the grammar file mechanism (similarly to BOM - [Listing 4](#) - and MasterLib - [Listing 10](#) - imports) in order to be processed and shown in the final report.

The flow for the EOL coverage declaration analysis is reported at [Figure 91](#) and described below.

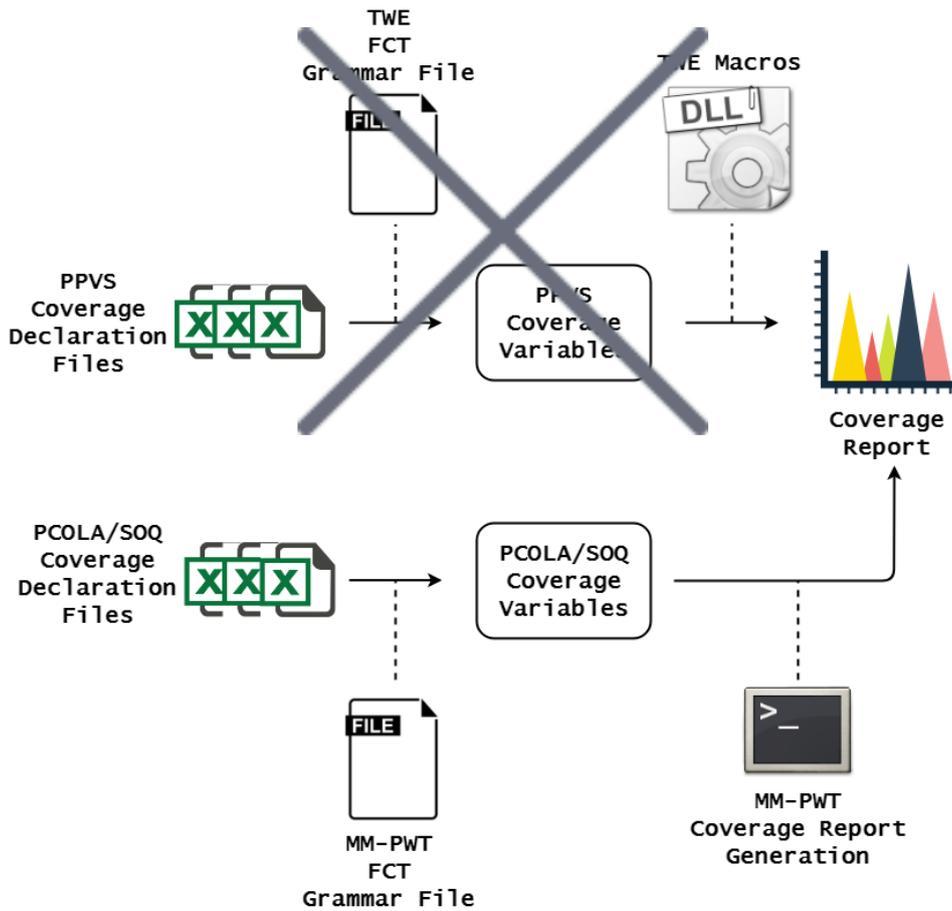


Figure 91: TWE - Customized Declaration Method Coverage Analysis Flow

Whenever a declaration method coverage process is launched, TWE imports the three files (in which coverage are declared) by using a grammar file. Natively, the import is based on PPVS because the final MACROS, that produce the report, work in PPVS. Due to the fact that these MACROS can not be customized, a new grammar file, capable of importing PCOLA/SOQ variables, has been introduced so that the proprietary script processes those variable to produce the final coverage report.

The customization concerning the EOL coverage declaration analysis is related to the **Declaration Files Import**. First of all, the structure of these files is described in order to finally define the customized grammar file.

- **Step File:** It is the file through which the Test Engineer declares the list of all test steps within the sequence. The structure is reported in [Table 26](#).

<i>Step Number</i>	<i>Step Name</i>	<i>Step Description</i>
1	fct_c12_c104	"Parallel Capacitors"

Table 26: TWE Customization - Declaration Method Step File

The *Step Number* attribute represents an ID for the step, whose name is reported in the *Step Name* field. In addition, a brief *Description* for each test is allowed.

- **Part File:** It is the file through which the Test Engineer declares the component PCOLA coverage values corresponding to each test step. The structure is reported in [Table 27](#).

<i>Ref</i>	<i>Step Number</i>	<i>Presence</i>	<i>Correct</i>	<i>Polarity</i>	<i>Live</i>	<i>Alignment</i>
C104	1	100.0	100.0	IGNORED	100.0	0.0

Table 27: TWE Customization - Declaration Method Part File

The component *Ref* is associated to the *Step Number* in order to specify the *Presence*, *Correct*, *Polarity (Orientation)*, *Live* and *Alignment* coverage values.

- **Pin File:** It is the file through which the Test Engineer declares the pin SOQ coverage values corresponding to each test step. The structure is reported in [Table 28](#).

<i>Ref</i>	<i>Step Number</i>	<i>Short</i>	<i>Open</i>	<i>Quality</i>
C104-1	1	100.0	100.0	0.0

Table 28: TWE Customization - Declaration Method Pin File

The pin *Ref* is associated to the *Step Number* in order to specify the *Short*, *Open* and *Quality* coverage values.

- **Grammar File:** Now that the structure of the three files is clear, the grammar file capable of importing the three files info is explained. It is a complex file which is capable of importing the following variables at test program ([Listing 28](#)), component ([Listing 29](#)) and pin ([Listing 30](#)) levels.

```
%teststep%      ! Step ID
```

Listing 28: TWE Customization - FCT Test Program Variables

```

%coverage_presence_fct%      ! PRESENCE
%coverage_correct_fct%      ! CORRECT
%coverage_polarity_fct%    ! ORIENTATION
%coverage_live_fct%        ! LIVE
%coverage_alignment_fct%   ! ALIGNMENT
    
```

Listing 29: TWE Customization - FCT Component Variables

```

%coverage_short_fct%        ! SHORT
%coverage_open_fct%        ! OPEN
%coverage_quality_fct%     ! QUALITY
    
```

Listing 30: TWE Customization - FCT Pin Variables

For any test step, the listed PCOLA/SOQ variables concerning FCT are imported.

9.7 Coverage Reports

Reports are the outputs of the entire flow, in which coverage results related to the adopted test line are resumed. The native TWE reports creation process is customized accordingly to MM-PWT desires. The idea behind such customization is reported in [Figure 92](#) and described below.

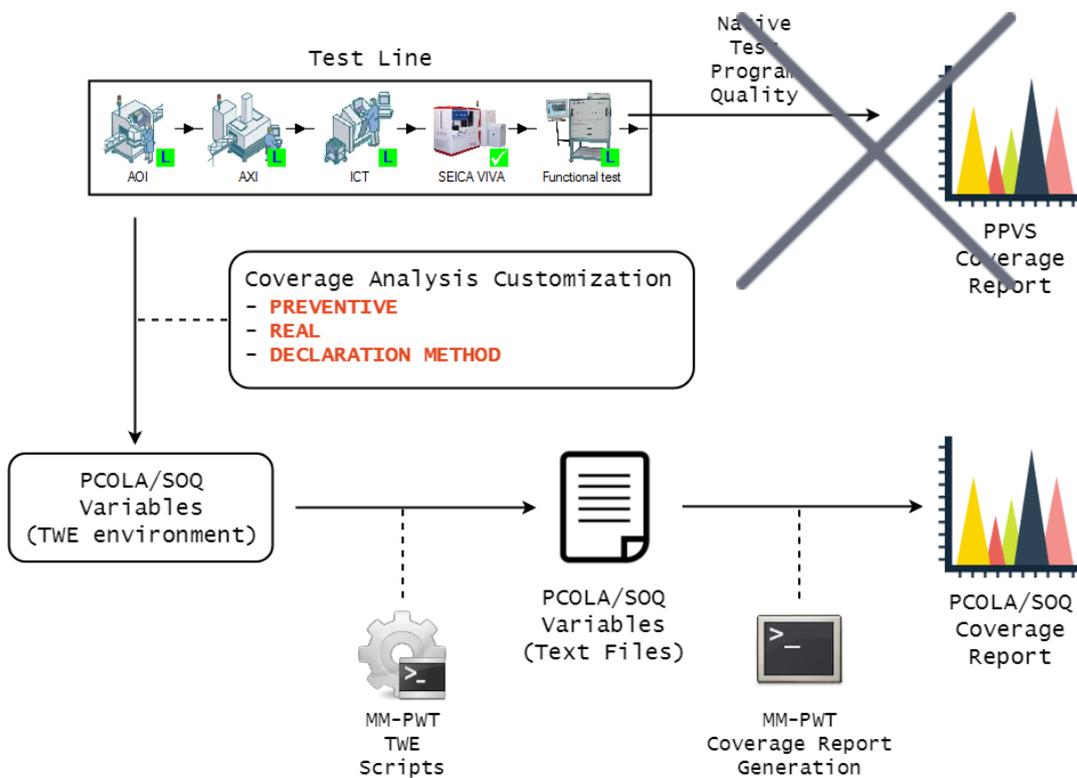


Figure 92: TWE - Customized Coverage Report Creation Flow

The native *Test Program Quality Analysis* would produce the PPVS Coverage Report. According to the customization previously described, concerning the production of PCOLA/SOQ coverage values (Preventive, Real and Declaration Method), some MM-PWT TWE Scripts are used to write down these values into text files such that they can be then processed by the MM-PWT proprietary script for the PCOLA/SOQ Coverage Reports generation.

The PCOLA/SOQ Variables Text Files generation is allowed by the **MM-PWT TWE Scripts** (whose customization is reported at [subsection 9.7.1](#)). These files are then manipulated by the **MM-PWT Coverage Report Generation script** in order to produce graphical ([subsection 9.7.2](#)) and numerical ([subsection 9.7.3](#)) reports.

9.7.1 MM-PWT TWE SCRIPTS

The mean used to let TWE communicate information to the MM-PWT proprietary script (developed in C++) is the *"Text File"*. TWE allows the user to develop Scripts with a particular syntax. In this case, the scripts are developed such that producing text files containing coverage information related to the adopted test line.

9.7.1.1 MM_PWT_test_line.scr

It is the script which is invoked at the end of the Test Program Quality Analysis in order to produce a .CSV file containing the list of the strategies involved in the test line. The script file is reported at [Listing 31](#).

```
%file%  
WRK:MM_PWT_TEST_LINE.csv  
%end%  
  
%text%  
%test_strategy%  
%end%
```

Listing 31: TWE Customization - *MM_PWT_test_line.scr* file

The macro **%file%** is used to manage an output file (.csv). The macro **%text%** is then used to write the test strategy into that file (a TWE macro is used for this purpose: **%test_strategy%**). An example of file content could be:

AOI , ICT , FCT

In order for this file to be produced whenever a Test Program Quality Analysis is performed, it is necessary to link the scripts to the .AA TWE project, as reported in [Listing 32](#).

```
[DfTChecker_Scripts]
Count=11
! . . .
SCRIPT6=1,VISAR00T\Data\MARELLI\CUSTOM\MM_PWT_TEST_LINE.SCR
! . . .
```

Listing 32: TWE Customization - Link Test Line Script to Project

9.7.1.2 MM_PWT_coverage_<strategy>.scr

It is the script, one per test strategy, which is invoked at the end of the Test Program Quality Analysis in order to produce a .CSV file containing the PCOLA/SOQ coverage values achieved by the related test strategy. The script file is reported at [Listing 33](#) (in the particular case of the ICT preventive technique, but it can be extended to any other).

```
%file%
WRK:MM_PWT_coverage_ICT.csv
%end%

%text%
%boardname% %coverage_access(ICT)%
%foreach device(mounted == 1)%
@ %ref%
    %function%
    %partnumber%
    %pin_cnt%
    %coverage_presence_ict%
    %coverage_correct_ict%
    %coverage_polarity_ict%
    %coverage_live_ict%
    %coverage_alignment_ict%
    %comment_ict%

    %foreach pin%
        %pin%
        %nodename%
        %coverage_short_ict%
        %coverage_open_ict%
        %coverage_quality_ict%
        %pintype%
        %comment_ict%
    %end%
%end%
%end%
```

Listing 33: TWE Customization - *MM_PWT_coverage_ICT.scr* file

Similarly of what described in [subsubsection 9.7.1.1](#), PCOLA/SOQ coverage values, for each mounted device, are written into the .csv file.

In order for these files to be produced whenever a Test Program Quality Analysis is performed, it is necessary to link the scripts to the .AA TWE project. For this purpose, the section reported in [Listing 34](#) must be added to the project .aa file.

```
[DfTChecker_Scripts]
Count=11
! . . .
SCRIPT0=1,VISAR00T\Data\MARELLI\CUSTOM\MM_PWT_coverage_AOI.SCR
SCRIPT1=1,VISAR00T\Data\MARELLI\CUSTOM\MM_PWT_coverage_AXI.SCR
SCRIPT2=1,VISAR00T\Data\MARELLI\CUSTOM\MM_PWT_coverage_ICT.SCR
SCRIPT3=1,VISAR00T\Data\MARELLI\CUSTOM\MM_PWT_coverage_FCT.SCR
SCRIPT4=1,VISAR00T\Data\MARELLI\CUSTOM\MM_PWT_coverage_VIVA.SCR
! . . .
```

Listing 34: TWE Customization - Link Coverage Scripts to Project

9.7.1.3 print_uncoveredPins_ICT.scr

It is the script which is invoked at the end of the ICT Preventive Coverage Analysis in order to produce a .CSV file containing a list of all uncovered pins for what concerns either *Open* or *Short* defect. The produced document can be a starting point for writing the Functional Test Program EOL (whose sequence can begin covering those pins not covered by ICT). The script file is reported at [Listing 35](#) (in the particular case of the ICT preventive technique, but it can be extended to any other).

```
%file%
WRK:ALL_UNCOVERED_PINS_ICT.csv
%end%

%text%
%boardname% - All Uncovered PINS (Preventive ICT)

%foreach device (mounted == 1 && hasPinUncovered == "true")%

    %foreach pin (pinsToPrint == "true")%
    %pin:10% %coverage_short_ict:7% %coverage_open_ict:7% %comment_ict%
    %end%

%end%

%end%
```

Listing 35: TWE Customization - print_uncoveredPins_ICT.scr file

For all devices with at least one uncovered pin (**hasPinUncovered** property), the *Open* and *Short* coverage values of their uncovered pins are listed (**pinsToPrint** property).

The **hasPinUncovered** is a customized component property defined by the rule reported at [Listing 36](#). It is initially set to *False* for all components. Then, for those components with either a pin with Short or Open uncovered, is set to *True*.

```
Rule Pins_not_Covered_ICT :

Assign criteria (hasPinUncovered = "false") to Device when:
  Device * is reported.

Assign criteria (hasPinUncovered = "true") to Device when:
  Pin criteria (coverage_short_ict == 0.0 || coverage_open_ict == 0.0) of Device
  is reported.
```

Listing 36: TWE Customization - *Component with uncovered Pin Rule*

The **pinsToPrint** is a customized pin property defined by the rule reported at [Listing 37](#). It is initially set to *False* for all pins. Then, for those pins with either Short or Open uncovered, is set to *True*.

```
Rule Pins_not_Covered_ICT :

Assign criteria (pinsToPrint = "false") to Pin when:
  Pin * is reported.

Assign criteria (pinsToPrint = "true") to Pin when:
  Pin criteria (coverage_short_ict == 0.0 || coverage_open_ict == 0.0) of Device
  is reported.

Assign criteria (pinsToPrint = "false") to Pin when:
  Pin (NC) of Device
  is reported.
```

Listing 37: TWE Customization - *Uncovered Pin Rule*

Both rules are contained into the **MM_PWT_coverage_custom.rul** file.

In order for these files to be produced whenever a Test Program Quality Analysis is performed, it is necessary to link scripts and rules to the *.AA* TWE project. For this purpose, the section reported in [Listing 38](#) must be added to the project *.aa* file.

```

[DfTChecker_Rules]
Count=3
! . . .
RULE1=1,VISAR00T>Data\MARELLI\CUSTOM\MM_PWT_coverage_custom.rul
! . . .

[DfTChecker_Scripts]
Count=11
! . . .
SCRIPT8=1,VISAR00T>Data\MARELLI\CUSTOM\print_UncoveredPins_ICT.SCR
SCRIPT9=1,VISAR00T>Data\MARELLI\CUSTOM\print_ICs_UncoveredPins_ICT.SCR
SCRIPT10=1,VISAR00T>Data\MARELLI\CUSTOM\print_MICRO_UncoveredPins_ICT.SCR

```

Listing 38: TWE Customization - Link Uncovered Pins ICT Scripts to Project

9.7.2 GRAPHICAL REPORT

Graphical Report (one of the added value of the Coverage Analysis offered by TWE, [subsection 7.2.4](#)) allows to visualize on the Layout view (in the form of a color code) the coverage values of the various strategies concerning the most significant PCOLA/SOQ variables (for instance, it does not make any sense showing *Quality* coverage for ICT since it would be always at 0.0%).

For this purpose, the introduced customization is summarized in [Figure 93](#) and described below.

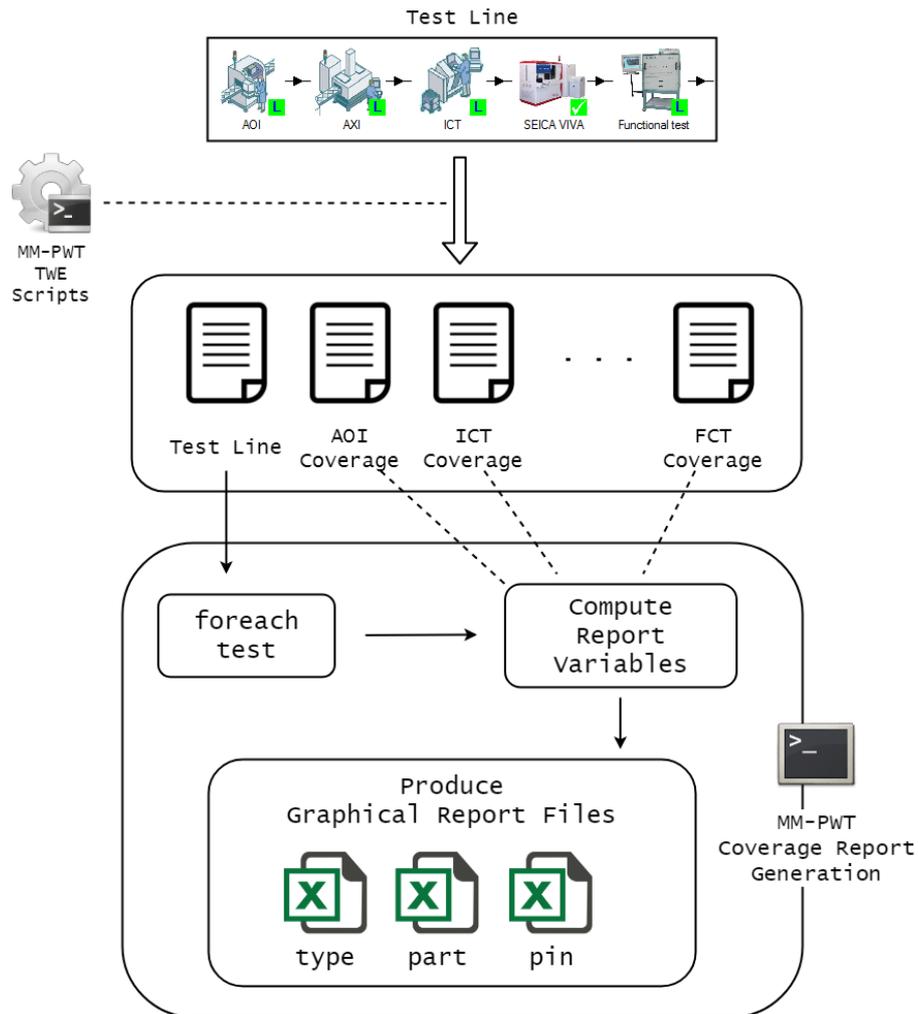


Figure 93: TWE Customization - Graphical Report Creation Flow

Basically, the script acquires the adopted test line (file produced by [subsection 9.7.1.1](#)) and for any strategy, processes the PCOLA/SOQ coverage text files (produced by [subsection 9.7.1.2](#)) in order to produce three files. These, in turn, are interpreted by TWE to produce the graphical report. In particular, the variables listed within the **TYPE** file (summarized at [Listing 39](#)) are visualized at component (taken from **PART** file) and pin (taken from **PIN** file) levels.

```

! ICT Accessibility
COVERAGE_NAIL

! AOI
COVERAGE(AOI)           ! COMPONENT   (all PCOLA/SOQ properties)
COVERAGE_QUALITY(AOI)  ! PIN        (Quality property)

! ICT
COVERAGE(ICT)           ! COMPONENT   (all PCOLA/SOQ properties)
COVERAGE_SHORT(ICT)    ! PIN        (Short property)
COVERAGE_OPEN(ICT)     ! PIN        (Open property)

! FCT
COVERAGE(FCT)           ! COMPONENT   (all PCOLA/SOQ properties)
COVERAGE_PRESENCE(FCT) ! COMPONENT   (Presence property)
COVERAGE_SHORT(FCT)    ! PIN        (Short property)
COVERAGE_OPEN(FCT)     ! PIN        (Open property)

! TEST LINE
COVERAGE(COMBINED)     ! COMPONENT   (all PCOLA/SOQ properties)

```

Listing 39: TWE Customization - Graphical Report Variables

9.7.3 NUMERICAL REPORT

The process used to generate the customized numerical Report is analogue to the one adopted for the Graphical Report (consult [subsection 9.7.2](#) to have more details). The difference is given by the fact that a web-based report is created in this case (HTML, CSS and JS files are generated instead of ".CSV"). The structure of such report presents both component and pin levels PCOLA/SOQ coverages (an example in [Figure 115](#) shows a pin level report).

The web-based files, interpreted by TWE to show the coverage report, are generated by the MM-PWT proprietary script by processing the coverage values from the text files produced by the MM-PWT TWE Scripts ([subsection 9.7.1.2](#)).

Part V

CONCLUSION

CONCLUSIONS

This chapter focuses on the results obtained at the end of the whole experience. A list of the achieved and missed goals is proposed in [section 10.1](#) in order to conclude with a brief anticipation of the possible future directions that the customized tool can take within Magneti Marelli Powertrain Testing Team ([section 10.2](#)).

10.1 Achieved and Missed Goals

This section provides an overview of the main points which have been achieved ([subsection 10.1.1](#)) and those which have not ([subsection 10.1.2](#)).

10.1.1 ACHIEVED GOALS

Repeatability is the most important achieved goal which encloses, in turn, several concepts. After this experience, in fact, the hardware fault coverage flow (passing through layout and design for test validation up to the coverage estimation) can be continuously repeated allowing eventual corrective-actions to be taken. *Repeatability* is intended as the **capability of the tool to provide fast analysis in terms of board classification, design/layout for testability validation and coverage estimation**. The added value brought from a quick coverage estimation avoid eventual wasting of time and human errors. In addition, design/layout for testability validations allow a closing loop cooperation between project development and testing teams. The old manual method for the coverage document generation as well as the proprietary metric, have been abandoned to leave place to an automatic flow working in PCOLA/SOQ (a worldwide standard for modeling component defects).

In other words, this experience produced a consistent milestone in the automation of the MM-PWT hardware fault coverage flow, from which future improvements and integration can take place ([section 10.2](#)). A deep confidence of the tool and debugging activity have been consolidated as well as a consistent number of introduced customization useful to adapt the tool to the company needs. However, some gaps have still to be filled before considering TestWay Express a stable product to rely on.

10.1.2 MISSED GOALS

The principal missed goal is related to the coverage computation, in particular for what concerns the **Real Coverage Analysis**. In [subsection 5.2.4](#), the MM-PWT de-

cision to adopt PCOLA/SOQ as defect-metric is justified. However, even though TestWay Express is presented by Aster as a tool capable of working in PCOLA/SOQ, technical analysis performed during the experience did not bring to the same assumption. For this purpose, some customizable parts have been properly modified in order to adapt the tool to work with the desired metric (as described for what concerns *Preventive Coverage Analysis* in subsection 9.6.1 and *Declaration Method Coverage Analysis* in subsection 9.6.3). However, the tool presents also some not customizable parts (such as Imports, for real coverage analysis, and Macros). Differently than Macros, which have been replaced with a proprietary script developed in C++, the effort needed to replace the Imports with an artificial mechanism has been considered excessive. A collaboration with Aster (producer of the tool) is taking place to fix the problem.

Everything is summarized in Figure 94, in which the customized phases are signed with a green circle. The not customized TWE imports are enough for the whole Real Coverage Analysis Analysis flow to be interrupted (no PCOLA/SOQ real coverage analysis can be performed).

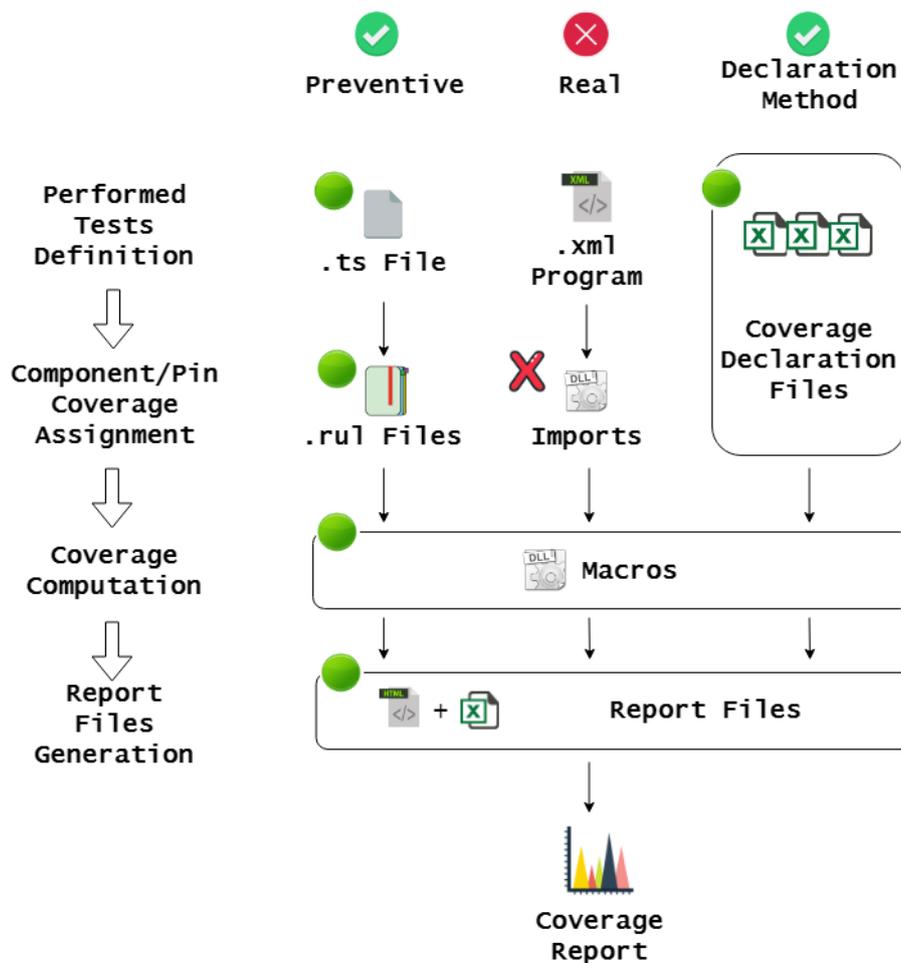


Figure 94: TWE - The whole Coverage Analysis Flow

10.2 Future of the Automatic Flow in MM-PWT

To conclude, a personal idea of the multiple directions that the project can take is proposed in this section.

- **Debugging & Tuning:** The customization process has been applied taking into consideration a single board. It is more opportune working with various projects in order to address eventual problems and properly tune the tool.
- **Native PCOLA/SOQ:** A future release of the tool working entirely in PCOLA/SOQ would additionally automatize the coverage flow by avoiding the use of the proprietary MM-PWT scripts (developed in C++ and not invocable from TWE environment) and other inaccurate customization.
- **Safety-Relevant Aspects:** With the increased incoming of safety-relevant aspects it would be reasonable to integrate the possibility to separately manage those components considered critical from the regulation ISO 26262.
- **EOL Feature:** As anticipated, TWE requires coverage declaration for what concerns Functional Test EOL. It is important to point out that the effort that this process requires is not negligible from the Test Engineer point of view that should, meanwhile writing the Test Sequence, declares also the coverage values with a particular syntax in order for them to be imported into TWE. It would be useful having a plugin to be used within TWE that allows the Test Engineer to easily declare coverage values.
- **Easier Use:** Integration and improvements can be added in order for the tool to be used by a common operator (completely unaware of the hardware coverage flow aspects) to produce certain data.

Part VI

APPENDIX

USER MANUAL

This Appendix provides a TestWay Express User Manual containing all essential information for the user to make full use of the tool during the entire Hardware Fault Coverage flow. In [section A.1](#), an *"How To"* for any main step of the flow is proposed.

A.1 How To

In this section the user is instructed to:

1. Select the step to be performed (moving to the related Section).
2. Verify the correctness of the PRE-CONDITIONS.
3. Follow the specified PROCEDURE.
4. Check the consistency of the obtained results (summarized in the POST-CONDITIONS).

In [Figure 95](#) an overview of the entire Hardware Fault Coverage Flow is proposed. An explanatory section is then dedicated for any block.

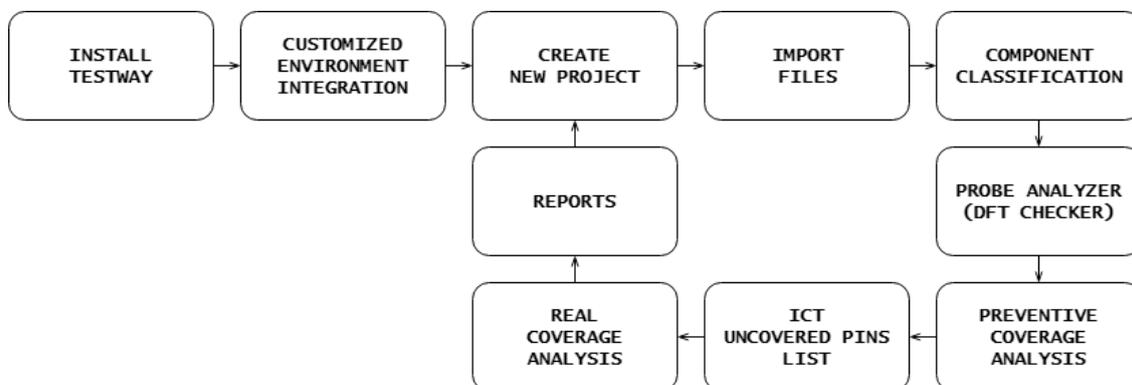


Figure 95: Hardware Fault Coverage - User Manual Flow

A.1.1 INTEGRATE THE CUSTOMIZED ENVIRONMENT

Once TWE has been installed, the MM-PWT customized environment has to be integrated. Through this step all customization, adopted to make TWE suitable for MM-PWT purposes, are made available for future uses of the tool.

PRE-CONDITIONS

- **TWE must have been installed:** The tool must be installed before proceeding.

PROCEDURE

1. Click [here](#)¹ to download **MARELLI.zip**.
2. Unzip **MARELLI.zip** into **C:/TestWay/Data/** using the password "*testway*". The unzipped **MARELLI/** folder should be got (it contains PROJECTS, CUSTOM, LIB1 and MODELS folders as well as two proprietary scripts).
3. Launch **MARELLI/prepareCustomizedEnvironment.exe** and follow the instruction. At the end, the message "*Customized environment has been loaded properly*" should be got.

POST-CONDITIONS

- **MM-PWT Customized Environment is now integrated into TWE:** The just integrated customized environment should be now ready to be used. To verify the correctness of the performed procedure, launch TestWay Express and make sure that the black-circle reported in [Figure 96](#) appears in the main window.



Figure 96: TWE MM-PWT Customized Environment - Main Window

NOTES

- This step must be performed only once (after TWE installation), as reported in [Figure 95](#).

¹ https://github.com/lucaledda/MM_TWE/raw/download/MARELLI.zip

A.1.2 CREATE NEW PROJECT

Whenever a new project has to be created, a new directory must be prepared. This step describes how to create a new project environment.

PRE-CONDITIONS

- **MM-PWT Customized Environment must have been integrated:** To integrate the customized environment, if not already done, consult [subsection A.1.1](#).

PROCEDURE

1. Launch **MARELLI/createNewProject.exe** and follow the instruction. At the end, the message *"The new project has been created"* should be got.

POST-CONDITIONS

- **New Project Directory:** The new project environment has been created. It is possible to check the **project Directory** presence within:

C:/TestWay/Data/MARELLI/PROJECTS/

("C:" or "D:" depending on the TWE installation root).

NOTES

- Note that only the project directory has been prepared in this step. The project characterization takes place in the next phases.

A.1.3 IMPORT FILES

BOM and CAD files contain all information necessary for a new project to be characterized. In this step it is explained how to import these files into an already prepared new project environment.

PRE-CONDITIONS

- **Project Directory must have been created:** The environment of the project must be present before proceeding with the files import. To create the new project directory, if not already done, consult [subsection A.1.2](#).
- **BOM FUJI:** The BOM file in FUJI format must be available.
- **CAD:** The CAD file in CAD format must be available.

PROCEDURE

1. BOM file

- a. Copy the BOM file (renamed **fuji.txt**) into **projDir²/bom/**.
- b. Launch **projDir/bom/from_FUJI_to_TWE.exe** to convert the BOM FUJI into a BOM importable by TestWay (check that, at the end of the script execution, the file **bom.txt** within **projDir/bom/** is created).

2. CAD file

- a. Copy the CAD file (renamed **pcb.cad**) into **projDir/cad/**.

POST-CONDITIONS

- **Characterized Project:** The project environment has been characterized with BOM and CAD files information. Check that the import has been successful:
 - a. Launch TestWay Express.
 - b. From the top bar, **File -> Open project** and select **projDir/projName.aa**. The project is loaded into the TWE environment.

² projDir : C:/TestWay/Data/MARELLI/PROJECTS/<project_under_analysis>/

- c. The layout view should appear. Click on different components in that view and check that related information on the left are shown (for this purpose open the *information window* by clicking on ). A result similar to what reported in [Figure 97](#) should be obtained.

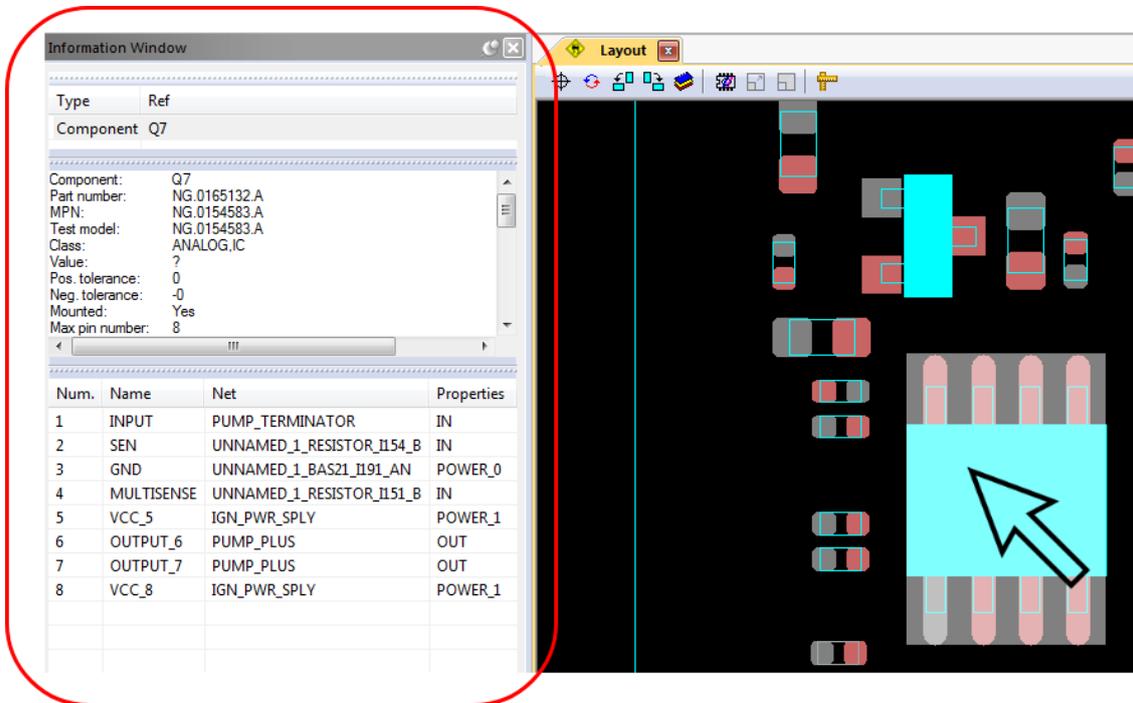


Figure 97: TWE Environment – Layout View Interaction

A.1.4 CLASSIFY THE COMPONENTS

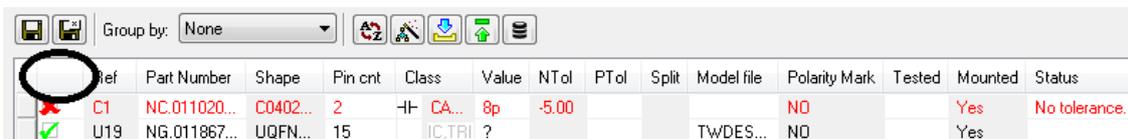
A full classification of components allows TWE to make better assumptions during the analysis. This is the reason why it is necessary to classify every component of the board before proceeding with later steps (DfT Checker, Coverage Analysis and so on so forth). Component Classification provides additional information to the ones already provided from BOM and CAD files.

PRE-CONDITIONS

- **BOM and CAD must have been imported:** The project must have been already created and characterized with both BOM and CAD files. To import such files, if not already done, consult [subsection A.1.3](#).

PROCEDURE

1. From the top bar, **Analysis -> Component Classifier**.
2. Click on the button black-highlighted in [Figure 98](#) to sort on top the components with missing information ("Group by" should be set at "None" to allow a single component classification).



Ref	Part Number	Shape	Pin cnt	Class	Value	NTol	PTol	Split	Model file	Polarity Mark	Tested	Mounted	Status
C1	NC.011020...	CD402...	2	HT	CA...	8p	-5.00			NO		Yes	No tolerance.
U19	NG.011867...	UQFN...	15		IC.TRI ?				TWDES...	NO		Yes	

Figure 98: Component Classifier - Missing Classification on Top

In this example C1 is not fully classified (red-marked). It is suggested to consult the column **Status** to get information about which properties are missing (in the figure, for example, "No tolerance" notifies that the tolerance property of C1 is missing).

3. Complete the classification until no components are red-marked. Some guidelines to follow are suggested:
 - a. **Fill up Missing Properties:** Firstly try to satisfy the suggestion given by the **Status** column (filling up missing information or updating bad data).
 - b. **Add Models:** Add the model files for ICs (those components with multiple pins).
 - i. Consult which are the ICs without a model. **Analysis -> DfT Checker**. From the window that opens, click on the "Custom Rules" tab and make sure that the following rule is selected:

VISAROOT:Data/MARELLI/CUSTOM/MM_PWT_DfT_custom.rul

At this point click on **Run**. A report is generated and shown. Expand the label "Missing models" to consult the list of ICs which still do not have an associated model, as shown in [Figure 99](#).

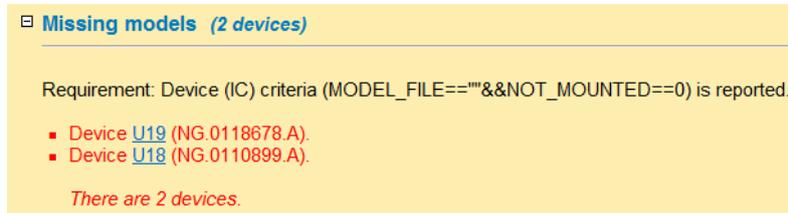


Figure 99: Component Classifier - Missing Models

- ii. Go back to the Component Classifier (**Analysis -> Component Classifier**).
- iii. Right click in the "Model file" cell field of the interested component and select **Generate model**, as shown in [Figure 100](#).

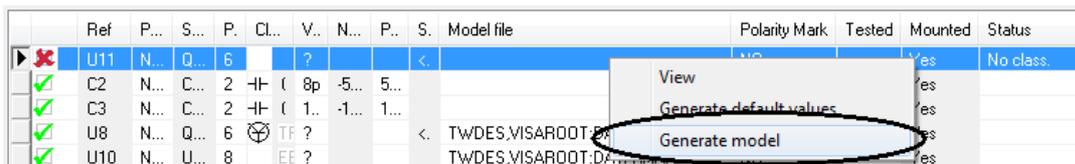


Figure 100: Component Classifier - Generate Model File

- iv. Right click on the same cell and select **Edit model**, as shown in [Figure 101](#).

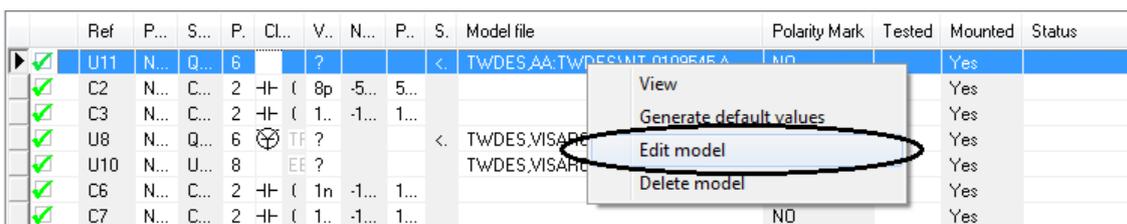


Figure 101: Component Classifier - Edit Model File

- v. Now click on the **Model Tab** to fill up essential information (the black-highlighted ones in [Figure 102](#)). In particular the boxes **Model** (with the Part Number of the component), **Class** (with the component classes), **DESC** (with a brief description of the component functionality) and **Type** (specifying, for each pin, whether it is IN, OUT, INOUT, GND or VDD).

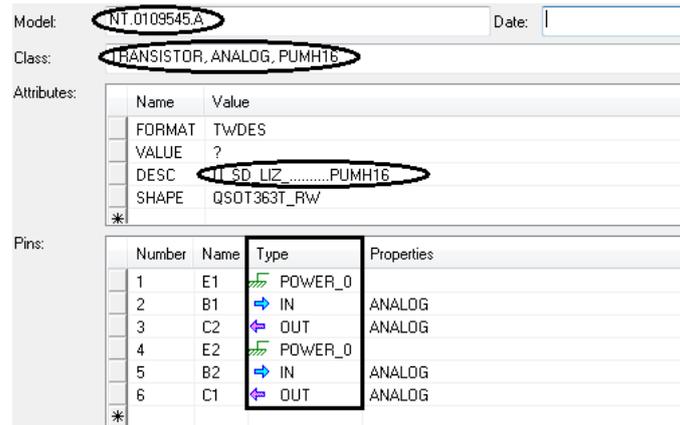


Figure 102: Component Classifier - Add Information Model

- vi. Repeat the steps from i. to v. for all ICs without a model (note that it is **important NOT TO SAVE** the component classifier status but only the new created models for the reason described at [section 9.4](#)).
- vii. Once all models have been created, copy them (contained within **projDir/TWDES/**) into:

C:/TestWay/Data/MARELLI/MODELS/

("C:" or "D:" depending on the TWE installation root).

- viii. Delete the folder **projDir/TWDES/**.
- ix. Click on the Component Wizard icon and **Identify** the **Model File** for **All parts**, as shown in [Figure 103](#).

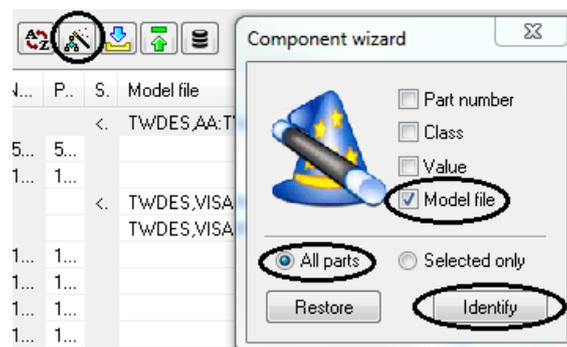


Figure 103: Component Classifier - Add Models from Model Library

4. Once that all components have been fully classified, click on the green arrow to update the new information into the Master Library, as shown in [Figure 104](#).



Figure 104: Component Classifier - Update MasterLib

POST-CONDITIONS

- **Fully Classified Project:** The Project is now fully classified. Further analysis could proceed.

A.1.5 LAUNCH PROBE ANALYZER (DFT CHECKER)

Design for Testability is the meeting point between project development and testing of a product. In this section, the user is instructed in the use of a design/layout checker to verify that the project is evolving accordingly to the constraints set up by the testing.

PRE-CONDITIONS

- **Fully Classified Project:** The project must be created ([subsection A.1.2](#)), characterized with BOM and CAD files ([subsection A.1.3](#)) and finally fully classified ([subsection A.1.4](#)).

PROCEDURE

1. From the top bar, **Analysis -> Probe Analyzer**. A window shows up, as reported in [Figure 105](#).

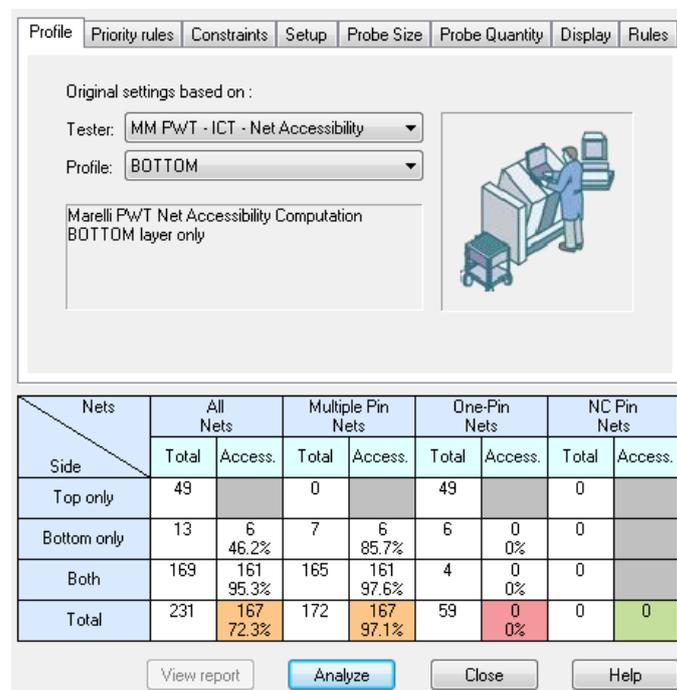


Figure 105: Probe Analyzer - Main Window

2. Select the **Tester** type.
Two customized profiles are available:
 - a) **MM PWT - ICT - Net Accessibility:** It provides to compute the percentage of accessible nets (the ones with at least one Test Point) by an ideal ICT profile. In this way it is possible to find out those nets without TP.

- b) **MM PWT - ICT - TPs Layout Checker:** It provides to compute the percentage of accessible nets by a real ICT profile (compliant to the MM-PWT layout rules). In this way it is possible to get all those violation in layout phase concerning Test Point placement.
3. Click on **Analyze**.
 4. Check the results by looking into the **Layout** (which can be also opened by clicking on the following icon: ). The inaccessible pins and TPs are red-marked whereas the accessible ones are green-marked (Figure 106). It is also possible to flip out the board by moving from Bottom side to Top side and vice-versa by clicking the **Side** icon:  .

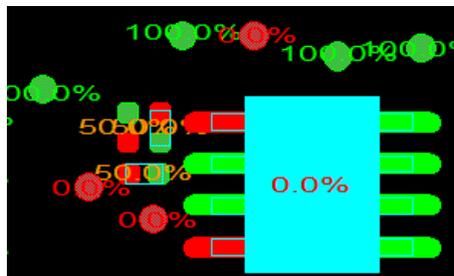


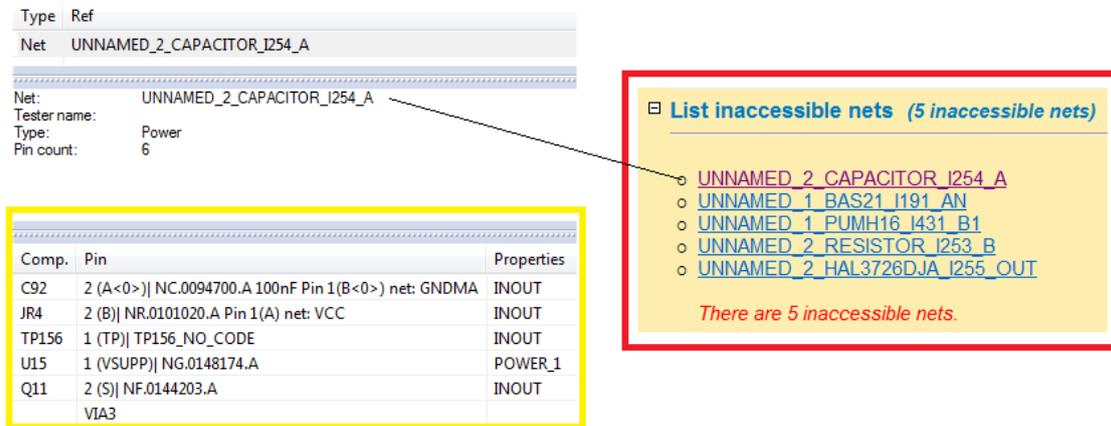
Figure 106: Probe Analyzer - Layout Tab, Accessibility Information

5. Check the results by clicking on **View report** (bottom-left button in Figure 105). Important information are:
 - a) **Probe Analyzer Configuration:** It provides a summary of the adopted Tester configuration. In the Figure 107 an example of a "Net Accessibility" Tester is shown up (all constraints are set to 0, meaning that the ideal ICT profile is capable of accessing to any TP).

Probe Analyzer configuration			
Constraints			
Mode	= Fixture		
Minimum Feature Size	= 0 mils		
Package Outline Clearance	= 0 mils		
Board Outline Clearance	= 0 mils		
Tooling Hole Clearance	= 0 mils		
Use solder mask	= NO		
Rules			
Object	Side	Size	Filter
TEST_POINT	BOTTOM	Default	
Probes			
Name	Nail Clearance	Package Clearance	Side
Default	0 mils	0 mils	Both

Figure 107: Probe Analyzer - Configuration Summary

- b) **List Inaccessible Nets:** It provides a list of all inaccessible multiple-pins nets (red-marked box in Figure 108). Open the **Information Window** (clicking on the icon ) and then click on the net-link to obtain all pins connected to that net (yellow-marked box in Figure 108).



The screenshot shows the Probe Analyzer interface. At the top, a table displays net details for 'UNNAMED_2_CAPACITOR_I254_A'. Below this, a table lists components connected to the net. To the right, a yellow box titled 'List inaccessible nets (5 inaccessible nets)' contains a list of five net names, with the first one highlighted. A red box highlights this list, and a red text message below it states 'There are 5 inaccessible nets.'

Type	Ref
Net	UNNAMED_2_CAPACITOR_I254_A

Net: UNNAMED_2_CAPACITOR_I254_A
 Tester name:
 Type: Power
 Pin count: 6

Comp.	Pin	Properties
C92	2 (A<0>) NC.0094700.A 100nF Pin 1 (B<0>) net: GNDMA	INOUT
JR4	2 (B) NR.0101020.A Pin 1 (A) net: VCC	INOUT
TP156	1 (TP) TP156_NO_CODE	INOUT
U15	1 (VSUPP) NG.0148174.A	POWER_1
Q11	2 (S) NF.0144203.A	INOUT
	VIA3	

List inaccessible nets (5 inaccessible nets)

- UNNAMED_2_CAPACITOR_I254_A
- UNNAMED_1_BAS21_I191_AN
- UNNAMED_1_PUMH16_I431_B1
- UNNAMED_2_RESISTOR_I253_B
- UNNAMED_2_HAL3726DJA_I255_OUT

There are 5 inaccessible nets.

Figure 108: Probe Analyzer – Inaccessible Nets List

POST-CONDITIONS

- **Design/Layout for Testability Information:** The information resulting from the Probe Analyzer can be used as feedback for project development phases to find a compromise which best fulfill the testing requirements.

A.1.6 GENERATE PREVENTIVE COVERAGE ANALYSIS

Preventive Coverage Analysis are estimation of what the test strategy is capable of covering. The analysis is based on a set of predefined rules. Unfortunately, it is very hard to classify the EOL Functional Test coverage within a set of rules. This is the reason why TWE offers Preventive Coverage Analysis for AOI, AXI and ICT only.

PRE-CONDITIONS

- **Launch Background Monitoring Script:** If not already done, launch the script:

projDir/PROJECTS/backgroundMonitoring.exe

This script runs in background continuously looking for new Coverage Analysis activities. Whenever a new activity is detected, it creates the MM-PWT Customized Report resuming coverage information of the just performed analysis.

- **Fully Classified Project:** The project must have been created ([subsection A.1.2](#)), characterized with BOM and CAD files ([subsection A.1.3](#)) and finally fully classified ([subsection A.1.4](#)). It is possible to check the quality of classification by clicking on **Analysis -> Test Program Quality**. On the window which shows up, as reported in [Figure 109](#), the **Classifier** indicator expresses the level of classification concerning the components of the board (the higher, the more accurate the coverage estimation).

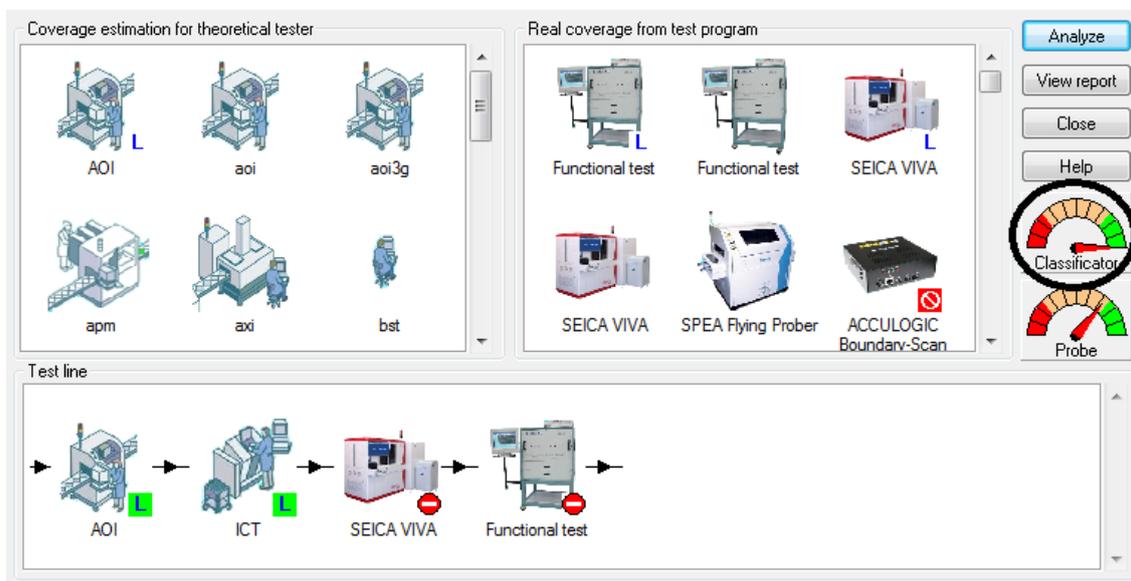


Figure 109: Test Program Quality - Component Classification Indicator

PROCEDURE

1. Click on **Analysis** -> **DfT Checker**. From the window that opens, click on the "Custom Rules" tab and make sure that the following rule is selected:

AA:strategy/reset.rul

At this point click on **Run**. Through this step, the TWE environment is prepared for a new coverage analysis (old PCOLA/SOQ variables are reset).

2. Compose the Test Line for which estimating the coverage. Test Line can be composed by a singular Test Strategy or by a combination of more. From the bottom-box "Test Line" reported in [Figure 109](#) it is possible to customize the proprietary line before launching the coverage analysis.

To **deselect** or **select** a test strategy: right click on the strategy and choose "Deselect/Select" (as reported in [Figure 110](#)).

All selected strategies will be taken in consideration when launching the coverage analysis. Note that in this phase the **Functional Test** must be **deselected** before proceeding (since no preventive EOL analysis are available).

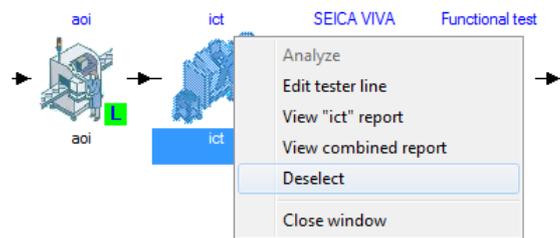


Figure 110: Test Program Quality - Deselect Test Strategy

3. Estimate the coverage by clicking on **Analyze** button (the one reported in the window shown in [Figure 109](#)).
4. Wait for the message "Waiting for the next activity." from the **Background Monitoring script**.
5. Go back to the TWE environment. An alert should appear ([Figure 111](#)).

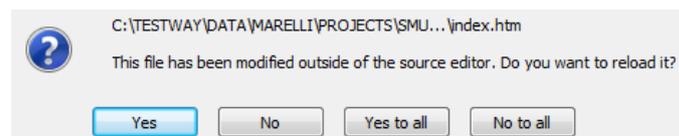


Figure 111: Test Program Quality - Refresh Report Alert

Click on **Yes** and then **Yes to all**. The PCOLA/SOQ report is now available.

- Consult the coverage report either of the entire line or specifically for a single strategy. Right click on the interested strategy (for example ICT) and click on **View "ICT" report** (as shown in [Figure 112](#)). Click on **View combined report** to consult coverage information related to the entire line.

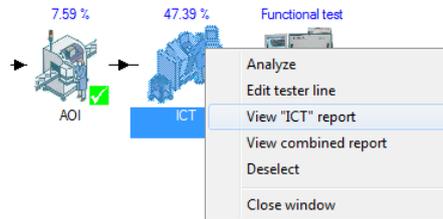


Figure 112: Test Program Quality – View report

- Consult also graphical coverage information by moving to the **Layout** tab and choosing one coverage information among all available by clicking on:  (for example two important information are **COVERAGE_SHORT(ICT)** and **COVERAGE_OPEN(ICT)** which highlight in which percentages the pins covered by In-Circuit Test).

POST-CONDITIONS

- **Customized PCOLA/SOQ Preventive Report:** It is possible to consult coverage information at Board, Component and Pin Level.

BOARD coverage information are reported in the main Test Report page ([Figure 113](#)). On the header right side, the "BOM COVERAGE" reports which is the percentage of covered "Presence" faults. Instead, the "FAULT COVERAGE" reports which is the weighted percentage of covered faults.

TEST REPORT FOR ICT 

BOARD NAME / NUMBER	SMU3_HW110	DATA PROCESSING REPORT	WRK:SMU3_HW110.htm
COMPONENT	241	BOM COVERAGE	87.9668%
NET ACCESS	74.89%	FAULT COVERAGE	59.01%

Figure 113: Test Program Quality – Board Coverage Info

COMPONENT coverage information are also reported in the main Test Report page ([Figure 114](#)) in which it is possible to scan all components consulting which faults are covered and which not.

COLOR		MEANING
		Ignored Defect
		High Coverage
		Medium Coverage
		Low Coverage
		Uncovered
		Not Mounted

REF.	PART NUMBER	PINS	P	C	O	L	A	S	O	Q
C90	NC.0094676.A	2	100.0%	100.0%		100.0%	0.0%	100%	100%	0%
C104	NC.0094676.A	2	100.0%	100.0%		100.0%	0.0%	100%	100%	0%
C31	NC.0094700.A	2	0.0%	0.0%		0.0%	0.0%	100%	0%	0%
C38	NC.0094700.A	2	0.0%	0.0%		0.0%	0.0%	100%	0%	0%
C44	NC.0094700.A	2	100.0%	100.0%		100.0%	0.0%	100%	100%	0%
C76	NC.0094700.A	2	0.0%	0.0%		0.0%	0.0%	100%	0%	0%
C77	NC.0094700.A	2	100.0%	100.0%		100.0%	0.0%	100%	100%	0%
C89	NC.0094700.A	2	100.0%	100.0%		100.0%	0.0%	100%	100%	0%
C96	NC.0094700.A	2	0.0%	0.0%		0.0%	0.0%	100%	0%	0%
C100	NC.0094700.A	2	100.0%	100.0%		100.0%	0.0%	100%	100%	0%
C261	NC.0094700.A	2	100.0%	100.0%		100.0%	0.0%	100%	100%	0%
C274	NC.0094700.A	2	100.0%	100.0%		100.0%	0.0%	100%	100%	0%

Figure 114: Test Program Quality - Component Coverage Info

PINS coverage information are available by clicking on the component REF link (left column of the Figure 114). The Figure 115 shows how these information appear. The **Component Scores** box reports coverage information related to the device (in all Presence, Correct, Orientation, Live and Alignment). The **Connection Scores** box reports summarized coverage information related to the pins of the device. The bottom box specifies coverage info of any device pin (in all Short, Open and Quality faults).

REPORT FOR REFERENCE DESIGNATOR "R107"

SUMMARY	
DEVICE FUNCTION	RESISTOR
PART NUMBER	NR.0123657.A
NUMBER OF PINS	2
COVERAGE SCORE	70%
COMMENT	COVERED PRESENCE (AT LEAST 1 OPEN PIN IS COVERED), COVERED CORRECT (RESISTOR TEST), IGNORED ORIENTATION (EQUIVALENT IF REVERSED), COVERED LIVE (FULL ACCESS)

"R107" COMPONENT SCORES					"R107" CONNECTION SCORES		
PRESENCE	CORRECT	ORIENTATION	LIVE	ALIGNMENT	SHORT	OPEN	QUALITY
100.0%	100.0%		100.0%	0.0%	100%	100%	0%

PIN NAME	NODE	PIN TYPE	SHORT	OPEN	QUALITY	COMMENT
1	VRS-IFA	INOUT	100.0%	100.0%	0.0%	OPEN COVERED BY LINK TEST, OPEN COVERED BY RESISTOR TEST, SHORT COVERED BY RESISTOR TEST, SHORT COVERED BY SHORT TEST
2	UNNAMED_2_CAPACITOR_I17_B	INOUT	100.0%	100.0%	0.0%	OPEN COVERED BY LINK TEST, OPEN COVERED BY RESISTOR TEST, SHORT COVERED BY RESISTOR TEST

Figure 115: Test Program Quality - Pin Coverage Info

A.1.7 GENERATE ICT UNCOVERED PINS LIST

After having generated the preventive coverage analysis, it is possible to generate a list of ICT uncovered pins (in particular those pins with either Short or Open uncovered faults).

PRE-CONDITIONS

- **ICT Preventive Coverage Analysis previously performed:** It is necessary that the procedure specified at [subsection A.1.6](#) has been performed for ICT before proceeding in this step. In this way, all coverage information concerning ICT are generated and a proprietary script can run to generate the desired list.

PROCEDURE

1. Click on **Analysis -> DfT Checker**. The window reported in [Figure 116](#) shows up.

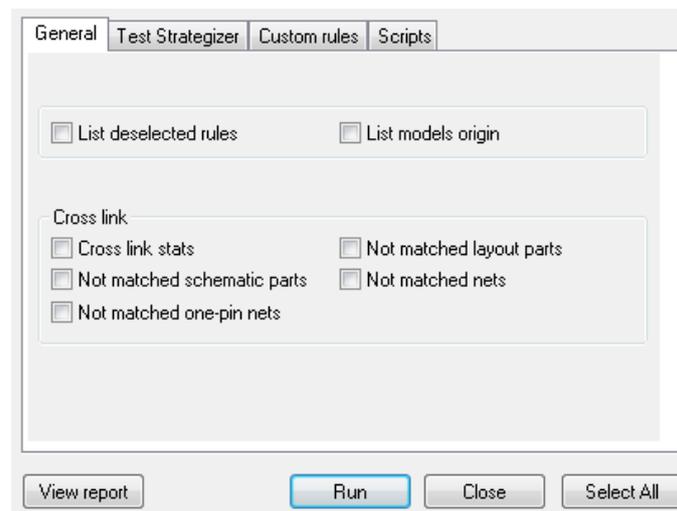


Figure 116: DfT Checker - Main Window

2. Make sure that on the Tab "Scripts" the following scripts are selected:
 - **VISAR00T>Data/MARELLI/CUSTOM/print_UncoveredPins_ICT.SCR**
 - **VISAR00T>Data/MARELLI/CUSTOM/print_ICs_UncoveredPins_ICT.SCR**
 - **VISAR00T>Data/MARELLI/CUSTOM/print_MICRO_UncoveredPins_ICT.SCR**
3. Click on **Run** to generate the script.

4. Consult the output Reports, found at the following paths:

- **projDir/DfTChecker/ALL_UNCOVERED_PINS_ICT.CSV**
- **projDir/DfTChecker/ICs_UNCOVERED_PINS_ICT.CSV**
- **projDir/DfTChecker/MICRO_UNCOVERED_PINS_ICT.CSV**

The structure of the scripts is the one reported in [Figure 117](#).

SMU3_HW110 - ICs Uncovered PINS (Preventive ICT)

PIN	SHORT	OPEN	COMMENT
U15-1	0	0.0	UNCOVERED SHORT AND OPEN (VDD PIN)
U15-2	0	0.0	UNCOVERED SHORT AND OPEN (GND PIN)
U15-3	0	0.0	UNCOVERED SHORT AND OPEN (GND PIN)
U15-4	100.0	0	OPEN UNCOVERED (POSSIBLE CLAMPING DIODE PARALLEL), SHORT COVERED BY SHORT TEST
U15-5	0	0.0	UNCOVERED SHORT AND OPEN (GND PIN)
U15-6	0	0.0	UNCOVERED SHORT AND OPEN (GND PIN)
U15-7	0	0.0	UNCOVERED SHORT AND OPEN (GND PIN)
U15-8	0	0.0	UNCOVERED SHORT AND OPEN (GND PIN)
Q6-1	0	0.0	UNCOVERED SHORT AND OPEN (GND PIN)
Q6-4	0	0.0	UNCOVERED SHORT AND OPEN (GND PIN)

Figure 117: Preventive ICT - Uncovered Pins List

POST-CONDITIONS

- **ICT Uncovered Pins List:** These lists can be useful as starting point for the EOL Test Program. In fact, it can begin from covering those pins which are not covered by ICT.

A.1.8 GENERATE REAL COVERAGE ANALYSIS

TWE offers even the possibility to generate Real Coverage Documents. It is important to notice that such generation is based on real report imports. Unfortunately, MM-PWT does not produce any AOI report during the flow and so no real coverage document can be generated for the Automated Optical Inspection technique. The following procedure explains, instead, how to generate real coverage document for both ICT and EOL strategies.

PRE-CONDITIONS

- **Fully Classified Project:** The project must be created ([subsection A.1.2](#)), characterized with BOM and CAD files ([subsection A.1.3](#)) and finally fully classified ([subsection A.1.4](#)).

A.1.8.1 ICT

The procedure for generating the EOL Real Coverage Document is based on the SEICA program import (in XML format). TWE elaborates such program in order to generate the resulting coverage document (mostly basing the analysis on the MACRO used during ICT). Unfortunately, as explained in [subsection 10.1.2](#), the current Real Coverage Analysis (based on imports) is available in PPVS only.

A.1.8.2 EOL

The procedure for generating the EOL Real Coverage Document is based on the **Declaration Method**³. The Test Engineer declares the component/pin coverage values for each test step. TWE elaborates such information in order to generate the resulting coverage document.

PROCEDURE: Precise procedure to be still defined. Anyway, a draft of the flow is proposed.

1. Develop the Test Program sequence by declaring component/pin coverage values related to each step. The syntax to be followed for the report generation is reported at [Listing 40](#).

³ It is role of the Test Engineer, while developing the test program, to DECLARE in which percentages components and pins are covered by any step of the test sequence.

```

! INPUTS POLARIZATION AND PWL OFF
O2SENSOR+1_POL [V] 2.432 Min 2.32 Max 2.52 P ! Test Step
@ C51_2_100_0_0_0_0 ! CompRef_NumPins_P_C_0_L_A
C51-1_100_100_0 ! PinRef_S_0_Q
C51-2_100_100_0 ! PinRef_S_0_Q

```

Listing 40: TWE Customization - EOL Declaration Report Syntax

Below each test, the component PCOLA coverage values (separated by "_") have to be defined preceded by the character "@". Then, line after line, the component pins SOQ coverage are declared as well.

2. Launch the script which takes in input the EOL Report and creates the three files (**FCT.part_tca.csv**, **FCT.pin_tca.csv**, **FCT.step_tca.csv**) importable by TWE.
3. Copy the three files into **projDir/Loader/FCT**.
4. Repeat same steps for generating Preventive Coverage (remembering to select **Functional Test** from the Test Line).

POST-CONDITIONS

- **Customized PCOLA/SOQ Final Report:** It is possible to consult coverage information at Board, Component and Pin Level:
 - **Board Level Information:** An example is shown at [Figure 113](#).
 - **Component Level Information:** An example is shown at [Figure 114](#).
 - **Pin Level Information:** An example is shown at [Figure 115](#).

A.1.9 GENERATE FINAL COVERAGE

The final coverage of the overall Test Line can be computed by selecting **AOI**, **SEICA VIVA** and **Functional Test** (as reported in [Figure 118](#)). The proposed Test Line is a mix of preventive and real profiles. As anticipated before, in fact, AOI is available in preventive analysis only. However, it is a good estimation of a real AOI profile.



Figure 118: Final Coverage - Best Test Line

BIBLIOGRAPHY

- [1] Bill Earl from Adafruit Industries. "Common Soldering Problems". URL: <https://learn.adafruit.com/adafruit-guide-excellent-soldering/common-problems>.
- [2] Carid. "What is a Fuel System 'Throttle Body'?" 2016. URL: <https://www.carid.com/articles/what-is-fuel-system-throttle-body.html>.
- [3] PCB Cart. "Printed Circuit Boards Assembly (PCBA) Process". URL: <https://www.pcbcart.com/article/content/pcb-assembly-process.html>.
- [4] Best Integration Technology Co. "AOI Light". URL: http://www.bestintech.com.tw/bestintech555/program_download/news/201507091512317013.pdf.
- [5] EDN. "Optimizing Printed Circuit Board Test Coverage". URL: <https://www.edn.com/Home/PrintView?contentItemId=4378216>.
- [6] Canadian Fuels. "ICE, HEV, PHEV and BEV – What they mean and what's under the hood". 2016. URL: <http://www.canadianfuels.ca/Blog/September-2016/ICE,-HEV,-PHEV-and-BEV-%E2%80%93-What-they-mean-and-what-s-under-the-hood/>.
- [7] MIS Electronics Inc. "PCB AOI Testing Service". URL: <https://www.miselectronics.com/PCB-AOI-Testing.html>.
- [8] National Instruments. "Archived: Detailed Calibration Procedures for Legacy Multifunction DAQ Boards". URL: <http://www.ni.com/tutorial/3241/en/#toc1>.
- [9] ASTER Technologies Lotz-Collins. "BOARD TEST - COVERAGE ANALYSIS. Is a board good because the test passes?" URL: https://aster-technologies.com/pub/en-ats2008-board_test_coverage_analysis.pdf.
- [10] Magneti Marelli. "Magneti Marelli's history 1918-2010". 2010. URL: <http://www.magnetimarelli.com/company/history/>.
- [11] Magneti Marelli. "Company". 2018. URL: <https://www.magnetimarelli.com/company>.
- [12] NEXLOGIC. "Functional Test: A Final Manufacturing Step". URL: <http://www.nexlogic.com/pcb-testing/functional-testing/>.
- [13] Paolo Nigro. "In-Circuit Test Techniques – One day internal training about ICT test phase". Magneti Marelli S.p.A. – Internal Course. 2018.
- [14] Electronics Notes. "Automatic optical inspection, AOI systems". URL: <https://www.electronics-notes.com/articles/test-methods/automatic-automated-test-ate/aoi-optical-inspection.php>.

- [15] Omron. "Q-upNavi". URL: <https://inspection.omron.eu/en/products/q-upnavi#images>.
- [16] SPEA. "The board tester SPEA 3030, over 500 copies". URL: <http://www.spea.com/Portals/0/SPEANewsTemplates/Generic.aspx?pItemId=227&pModuleId=844&pViewType=AllPress>.
- [17] Autodesk Sam Sattel. "Top 10 PCB Soldering Issues". URL: <https://www.autodesk.com/products/eagle/blog/top-10-soldering-issues-can-ruin-pcb-design/>.
- [18] Seica. "SEICA - About Us". URL: <https://www.seica.com/about-us/>.
- [19] C J Smith and Kaendorff. "ADVANTAGES AND LIMITATIONS OF AN AUTOMATED VISUAL INSPECTION SYSTEM". In: *A Journal of Industrial Engineering* 5.1 (1991), pp. 27–36.
- [20] ASTER Technologies. "ASTER's answer to industry 4.0". URL: <https://aster-technologies.com/en/>.
- [21] ASTER Technologies. "TestWay: Design for Test and Test Coverage Analyzer". URL: <https://aster-technologies.com/en/products/testway/>.
- [22] ASTER Technologies. "TestWay Express". URL: <https://aster-technologies.com/pub/en-twExpress-Brochure.pdf>.
- [23] ASTER Technologies. *TestWay Express - User Manual*. TWE - Official User Manual. 2018.
- [24] Keysight Technologies. "A New Process for Measuring and Displaying Board Test Coverage". URL: https://www.keysight.com/upload/cmc_upload/All/Apex_KParker_010903.pdf.
- [25] Wikipedia. "Automated optical inspection". URL: https://en.wikipedia.org/wiki/Automated_optical_inspection.
- [26] Wikipedia. "Ball grid array". URL: https://en.wikipedia.org/wiki/Ball_grid_array.
- [27] Wikipedia. "Burn-In". URL: <https://en.wikipedia.org/wiki/Burn-in>.
- [28] Wikipedia. "In-circuit test". URL: https://en.wikipedia.org/wiki/In-circuit_test.
- [29] Wikipedia. "Join (SQL)". URL: [https://en.wikipedia.org/wiki/Join_\(SQL\)](https://en.wikipedia.org/wiki/Join_(SQL)).
- [30] Wikipedia. "Part Number". URL: https://en.wikipedia.org/wiki/Part_number.
- [31] Wikipedia. "Powertrain". URL: <https://en.wikipedia.org/wiki/Powertrain>.
- [32] Wikipedia. "Printed circuit board". URL: https://en.wikipedia.org/wiki/Printed_circuit_board.
- [33] Wikipedia. "RC time constant". URL: https://en.wikipedia.org/wiki/RC_time_constant.

- [34] Wikipedia. "*Surface mount technology*". URL: https://en.wikipedia.org/wiki/Surface-mount_technology.
- [35] Wikipedia. "*Through-hole technology*". URL: https://en.wikipedia.org/wiki/Through-hole_technology.
- [36] Wikipedia. "*Via (electronics)*". URL: [https://en.wikipedia.org/wiki/Via_\(electronics\)](https://en.wikipedia.org/wiki/Via_(electronics)).
- [37] Yelo. "*Laser Diode Reliability Bathtub Curve*". URL: <https://www.yelo.co.uk/index.php/products/laser-diode-testing/laser-diode-life-test>.
- [38] Marantz Electronics Ltd. "*Mek 9 camera AOI technology for optimal visualisation of PCBs*". URL: <http://www.marantz-electronics.com/9-camera-aoi-technology/>.



RINGRAZIAMENTI

A conclusione di questo percorso non posso che ringraziare il Prof. Matteo Sonza Reorda e l'Ing. Paolo Nigro, per la grande opportunità offertami e il puntuale supporto durante tutta l'esperienza. In Magneti Marelli ho avuto il piacere di confrontarmi con un team esperto, compatto, dotato di un ricchissimo background tecnico ma soprattutto di un'immensa umanità. Una particolare menzione va agli Ing. Paolo Caruzzo e Roberto Saroglia, la cui professionalità mi ha fatto acquisire nuove motivazioni. In un mondo a me sconosciuto ho percepito pazienza e comprensibilità. Tutto ciò mi ha fatto sentire, nelle giuste misure, tanto coinvolto quanto responsabile.

Grazie agli amici di Torino. Grazie a Pane, Piergiovanni, Salvo, Giovanni e Danilo per aver reso questo percorso meno arduo. Grazie ad Antonio e Nacho, persone semplici che mi hanno saputo trasmettere la serenità nei giusti momenti. Grazie a Edoardo, compagno di mille progetti ma soprattutto persona umile e impeccabile da cui ho solo potuto prendere spunto.

Grazie agli amici di Carbonia. Mi sono sentito parte di un gruppo stupendo dove "O tutti o nessuno", in cui il significato di amicizia è ancora vero. Grazie ad Andrea, Giulia, Federico, Fabrizio, Davide Corsini, Davide Massa, Leonardo, Matteo, Simone, Riccardo, Martina e Nadia. Con voi ho vissuto dei momenti indescrivibili che porterò sempre con me.

Grazie ai *Coinquilini*, ragazzi dal cuore grande. Grazie alla compagnia di Alessio e Mattia. Mi avete sempre fatto sentire a casa, nel vero senso della parola. Grazie a Francesco, il ragazzo che mi sembra conoscere da una vita. L'Amico con la A maiuscola che farebbe di tutto per mettere gli altri al primo posto.

Grazie agli amici di sempre, quelli con cui sono cresciuto, coloro i quali mi hanno dato un'identità. Grazie a Luca, ragazzo tutto d'un pezzo di cui ho apprezzato semplicità e intraprendenza. Grazie a Simone, che ha sconfitto la sua cinicità mostrandomi sempre grandissimo affetto. Grazie a Giacomo, il bambino delle gite alle elementari diventato ormai un punto di riferimento. Grazie ad Andrea, per aver corrisposto fiducia e onestà. Per essere stato il mio esempio di ambizione e di amicizia. Per essere stato sempre se stesso, ciò che davvero apprezzo di lui.

Grazie a Laura, Beatrice e Giulia che hanno sempre preservato l'intesa di una vita fa. Grazie a Nicola, la colonna portante che mi ha spalleggiato in questa piccola ascesa caratteriale e professionale. E' stato per me l'esempio di aspirazione che, contro ogni difficoltà, si trasforma in concretezza. Grazie a Chiara, la ragazza che mi completa. Grazie per le lezioni obiettive, i consigli da vera amica e mai scontati. Grazie perchè ha sempre voluto esclusivamente la mia serenità. Lei, che è riuscita a tirare fuori tutto me stesso e adesso ne accetta ogni sfaccettatura.

Grazie ai nonni e il loro amore incondizionato. Grazie a Nonna Giovanna per aver tirato fuori il carattere nelle difficoltà. Sapeva che ci avrebbe potuto regalare ancora tanti sorrisi. Grazie a Nonno Elio per avermi fatto sentire importante come nessuno mai è riuscito a fare. Grazie a Nonna Verdina, che sicuramente sarà comoda da qualche parte per godersi questo momento. Sei stata in ogni mio silenzio, in ogni mio pensiero, in ogni mio post-esame quando, inconsciamente, prendevo il telefono per chiamarti.

Per concludere, grazie alle persone che mi hanno dedicato una vita intera. Grazie a Sara, per aver responsabilizzato il mio comportamento da fratello maggiore. Adesso è lei il modello da seguire. Grazie a Orazia, che con il giusto affetto mi ha guardato crescere e in sofferente silenzio ha accettato la mia lontananza. Avrebbe fatto questo e altro per vedermi realizzato. Grazie a Marco, che senza istruzioni è riuscito ad essere il Papà perfetto. Grazie a lui ho capito che nulla è dovuto, tutto va sudato e guadagnato. Dietro il suo "*darmi torto*" si nascondevano gli insegnamenti ad un bambino che, un giorno, sarebbe stato pronto a cavarsela da solo. Grazie Papà, un giorno vorrei diventare per qualcuno l'uomo che tu sei stato e che sarai per me. Ma per abbracciarti so che dovrò aspettare il prossimo gol del Cagliari.

Grazie Mamma e Papà, davvero. Grazie perchè avete costruito una vita intorno a me. Siete delle persone esemplari.