Automatic generation of in-circuit tests for board assembly defects

Citation for published version (APA):

van Schaaijk, H., Spierings, M., & Marinissen, E. J. (2018). Automatic generation of in-circuit tests for board assembly defects. In *Proceedings - 2018 23rd IEEE European Test Symposium, ETS 2018* Institute of Electrical and Electronics Engineers. <https://doi.org/10.1109/ETS.2018.8400714>

DOI:

[10.1109/ETS.2018.8400714](https://doi.org/10.1109/ETS.2018.8400714)

Document status and date:

Published: 29/05/2018

Document Version:

Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Automatic Generation of In-Circuit Tests for Board Assembly Defects

Harm van Schaaik^{1,2}Martien Spierings¹Erik Jan Marinissen^{2,3}¹ Prodrive TechnologiesScience Park Eindhoven 5501
5692 EM Eindhoven
The Netherlandsharm.van.schaaik@prodrive-technologies.com
martien.spierings@prodrive-technologies.com² Technische Universiteit EindhovenDen Dolech 2
5612 AZ Eindhoven
The Netherlands
h.a.h.v.schaaik@student.tue.nl
e.j.marinissen@tue.nl³ IMECKapeldreef 75
3001 Leuven
Belgium
erik.jan.marinissen@imec.be

Abstract — The components and the solder joints that are made during assembly to hold components to their printed circuit board can suffer from defects and therefore need to be tested. Many research papers on board-assembly testing focus on boundary scan test, processor-controlled test, or other powered digital testing techniques that mostly ignore the indispensable passive circuits and that can incur damage that could have been avoided by executing a non-powered test first. In-circuit testing is a non-powered test method that applies stimuli and measures responses using probe needles. However, often used self-learning solutions for designing these tests need a known-good-board, entailing significant disadvantages. In this paper, a software tool is described that automatically generates in-circuit tests based on the product design files, without requiring probe access on every net. Furthermore, the tool indicates where on the board fault coverage is not maximal, and hence where extra probe access will improve the test quality.

1 Introduction

In board assembly, electronic components are soldered onto a printed circuit board (PCB). Boards contain a mix of passive and active discrete components, as well as (digital) integrated circuits (ICs). We distinguish components based on through-hole technology (THT) and surface-mount technology (SMT). THT components, of which the leads are inserted into holes in the PCB, have been largely replaced by the much smaller SMT components which are placed on top of the board.

Board assembly suffers from defects, and hence the assembled PCBs need to be tested. The bare PCB as well as the components are assumed to be fault-free, typically achieved by manufacturing tests by their respective suppliers. Hence, what need to be tested after board assembly is whether or not the components are correctly soldered onto the PCB. Defects considered are expressed in the PCOLA-SOQ [1] score; this method assigns a coverage score to Presence, Correctness, Orientation, Liveliness, and Alignment properties of each device on the board and Short, Open, and Quality score to all the connections on the board.

Many research papers on board-assembly testing focus on “non-intrusive” testing techniques [2] such as boundary scan tests [3] or processor-controlled tests. However, such papers typically ignore that discrete components are often an indispensable part of a board design. Passive components can typically only be verified to be ‘present’ with such non-intrusive test techniques. Furthermore, these non-intrusive methods typically require the board to be powered and therefore might incur damage that could have been avoided by executing a non-powered test first.

In-Circuit Testing is a board-assembly test method in which components are electrically tested through probes of a bed-of-nails fixture that are connected via a switch matrix to a parametric test instrument. Ideally, every component-under-test (CUT) has individual probe access, and does not have false paths. If that is indeed the case, the CUT can simply be measured. However, this is not true in the reality of actual boards.

So-called “self-learning” solutions [4] characterize a board based on the impedance patterns between the available probe locations, and compare those to the impedance patterns obtained for a known-good (“golden”) board. Serious drawbacks of self-learning are the identification of a golden board and the fact that the test coverage of the various components on the board remains undetermined.

In this paper, we describe the first stage of the development of a software tool that automatically generates, based on a boards design netlist, in-circuit tests for board assembly defects of passive components. In this stage only passive components are considered. Support for other components will be implemented in later stages. We use graph-theoretic algorithms to identify probe locations and assignment of test signals to those probe locations such that (1) the granularity of the clusters is as small as possible, and (2) any alternative (“false”) paths between two target probe points are neutralized by guarding them with additional probes.

2 System Inputs

The test is generated based on the the design files of the device-under-test (DUT). The data that is needed is component information (e.g. reference designator (refdes [5]), type, value, tolerance) and interconnection data, i.e. the nets that describe the connections between the components on the board. Since the possibilities to test components are limited by the capabilities of the board tester, the tool needs to know the capabilities of the board tester, such as: (1) the voltage and frequency range of the signal generators, (2) the sensitivity and measurement ranges of the tester, (3) The impedance sweet spot in which the tester yields the most reliable tests, (4) the maximum guard ratio, i.e. the amount of current flowing in the guard path versus the amount of current flowing in the measurement path.

For further computation, we convert the netlist into an undirected multigraph $G = (V, E)$, where V denotes the set of vertices and E the set of edges. Contrary to what one perhaps intuitively would expect, we found that in our application representing components by edges and nets by vertices works best. However, multi-pin components, such as transistors and ICs, do not fit this representation, and therefore these components are represented by a vertex as well, with edges representing the pins of the component. In the graph net-vertices are marked to either have probe access or not. The edges are provided with a label that contains the refdes, component type, and value and tolerance.

Figure 1: Netlist handling example.

An example netlist and its corresponding graph are shown in Figure 1. Nets which have probe access are indicated as green vertices, while nets without probe access are shown as red vertices. The edge labels of the components are shown next to the corresponding edge.

3 Test Generation for Passive Components

After the graph is generated tests should be generated for the components in the graph. Since not every component can be tested in isolation, either due to components being in parallel or due to a lack of probe access, components will be grouped. In this paper these groups are defined as structures. A structure containing a single component is defined as an isolated structure, all other structures are defined as compound structures. Since tests will be generated for structures instead of components, this paper uses the term structure-under-test (SUT) instead of CUT. In order to generate tests for the structures, first a pre-processing step is performed on the graph that determines all the information that is needed to generate the test.

3.1 Graph Pre-Processing

Before tests will be mapped on the structures, the graph has to be pre-processed first. For each structure the following data needs to be decomposed: (1) which components are in the structure (2) which (conducting) paths are parallel to the structure i.e. the false paths. (see Section 3.1.2) and how to mitigate these, (3) the highest voltage that can safely be applied to a net.

Since this procedure focuses on generating tests for passive components, the active components are (temporarily) removed from the graph. Since passive components can be tested on a voltage low enough (e.g. 200mv) to not excite the PN junctions of active components, their influence is generally negligible. In order to group unreachable components into reachable structures, the graph is *collapsed*. Next the graph is split up into smaller subgraphs to lower the computational effort of the remaining part of the pre-processing steps. Next for all structures the minimal guard sets are determined, which describe all possible ways to mitigate the influence of the false paths. Finally the active components are considered again in order to determine the maximum voltage that can be used safely. These steps are explained in detail in Sections sections 3.1.1 to 3.1.3.

3.1.1 Graph Collapsing

The goal of the collapsing process is to combine unreachable isolated structures into reachable compound structures. It does by exhaustively combining edges that are in series or in parallel. Expanding the compound structures would again yield the original graph.

An example of the collapse process is shown in Figure 2. The graph of Figure 2(a) shows the original graph before collapsing. Figure 2(b) shows the graph in which the parallel edges are combined into a single structure. This process continues until there are neither edges in parallel nor unreachable edges in series. The result is shown in Figure 2(c).

Figure 2: Example of the collapsing process.

It might happen that a collapsed structure is still not reachable by probes on either ends (e.g. edges e_8 , e_9 and e_{10} of Figure 2(c)). If this is the case it can be concluded that probe access has to be added to reach the components within this structure.

3.1.2 Subgraphs

A structure is tested by connecting a source probe to one of the vertices of the structure, a measurement probe to the other, and executing a test routine. However, since the structures are tested while having been soldered onto a PCB, there might be paths other than the SUT that allow current to flow from source to measurement. These paths are called false paths.

Note that false paths can only be simple paths; a net cannot have multiple voltages at the same time and current cannot flow through it more than once. A closer look at these false paths reveals an interesting property, viz. that edges being in a false path of the SUT is both a reflexive and symmetric relation. If one observes that each false s, m -path of an edge $e = \{s, m\}$ together form a simple cycle, the rest of the proof is trivial. From this it follows that the connected components can be split into subgraphs that for each edge e contains e itself and all of its false paths. This split happens to be equal to the biconnected components [6] of the graph.

3.1.3 Minimal Guard Set

If a false path is not guarded it likely will influence the measurement. However, if too many nets are guarded in a false path, components are shorted between two guarded nets, which negatively influences the guard ratio. Moreover, since the impedance of these components might be dependent on the frequency, the ideal set of nets that are guarded is dependent on the selected frequency that will be used to test the structure.

The goal of the minimal guard set algorithm is to find all possible sets of vertices that guard all false paths with as few components shorted between guards as possible. It does this by first discovering all paths. Next all vertices without probe access are removed since these cannot be guarded. These paths are then sorted on length to prevent that a path gets guarded twice because it coincides with a shorter path. Finally the algorithm recurses the paths, selecting new nets to guard if the path wasn't already guarded by a previous selected net. All other vertices in the path are marked as 'not preferred' since selecting these would cause components to be shorted between guards.

3.1.4 Maximal Voltage

The maximum voltage that can be safely applied to a net is typically limited by the active components that can be reached from that net. As mentioned in Section 3.1 a maximal safe voltage can be assumed in order to prevent damage. However, if all paths between a source probe and active components are forced to a potential of $\sim 0V$, this is not necessary. Guard and measurement probes exactly do this. Therefore by discovering if there are any paths from the source probe to an active component, without crossing either an guarded or a measured vertex, the measurement can be determined to be limited by a safe voltage.

3.2 Test Mapping

Isolated structures contain only a single component, and therefore it is easy to create a library of tests that can be mapped on the isolated structures of the DUT to cover all the cases. However, for compound structures this is not true since these can consist of multiple different components, in multiple different combinations. To maximize the test coverage a generic test method is needed that tries to map a test on a structure if there is no test found in the test library. We deploy both solutions to get the best of both worlds.

3.2.1 Library Test Mapping

The library test mapping procedure maps tests from a pre-defined test library onto the structures of the DUT. This library contains product-independent tests, which all specify a so called *mold* which defines which combinations of components can be covered by the test, and a test mapping procedure.

Figure 3: Matching of a compound mold.

If an (expanded) structure and a mold are isomorphic [7] such that a mapping from the source and measurement probes to the reachable vertices of the structure is preserved, and all of the labels of the mold and structure match, then the mold matches and the mapping procedure will continue. Figure 3 shows an example of a compound mold. Only structure 2 matches since it contains the right component types, the combined value of the capacitors is within the specified range, while the orientation is irrelevant.

3.2.2 Generic Test Method

All structures that could not be covered by a test defined in the test library are handed to the generic test method (GTM). The generic test method is a method that tries to map a test as effective as possible, regardless of what is inside that structure. The generic test method can only perform impedance measurements and is therefore more limited than the tests in the library.

To determine the frequencies to test a structure at, the generic test method has to determine at which frequencies the components in the structure are *visible*. Just like the components inside the structure, the impedance of the structure itself has a tolerance. Using modified nodal analysis [8] the

currents through and voltages over each component can be determined. If the current through a component within the structure is large enough to be not within the tolerance of the structure, it is not current-dominated and its presence can be determined. Analogous, if a component is not voltage-dominated, it can be determined to be both correct and live.

Using this principle, the PCOLA-coverage can be determined for each of the frequencies by scanning over the frequency domain of the board tester. By selecting as few as possible frequencies that result in maximal coverage, the efficiency can be optimized. In order to determine the guard ratio, the impedance of the false path is determined next. Finally a suitable voltage is selected that is lower than the determined maximal voltage, but high enough for the board tester to perform a reliable measurement. From all these potential tests, all tests that are superfluous are removed.

Figure 4: Example of the generic test method.

Figure 4 shows an example of the generic test method applied on a structure consisting of a resistor and capacitor in series. Since the components are in series the current through both components is always 100% of the total current. The voltage over each of the components does depend on the frequency and therefore the resistor is voltage-dominated for low frequencies, while visible for higher frequencies. If the guard ratio for a given frequency is within the tolerable range of the board tester, a voltage can be selected. If in the example both the frequency and voltage are optimal somewhere in the middle of the range, both components have optimal coverage and the hardware of the tester is able to supply the current.

4 Experimental Results

We selected three boards for consumer and industrial applications from the Prodrive Technologies product catalog for our experiments: (1) the QA board is specially designed for qualification of the ICT tester hardware, (2) the EQDM board is designed towards maximum ICT access, (3) the ECU board is the first board that has a test running in production that is generated using this tool. For the largest board in this comparison, EQDM, that has over 3000 components, generating the test takes about 15 minutes on a PC with a Core i7-6820HQ CPU at 2.7 GHz and 8 GB of DDR4 RAM at 2.133 GHz.

The tests generated for the ECU board are executed using set of 15 boards. 252 of the 299 tests (84,28%) proved to be stable with a $C_{pk} \geq 2$ [9], 47 tests required a minor change to e.g., limits or frequency in order to get a stable test. The generated test for the QA board ran successfully, none of the tests steps proved to be unstable.

5 Conclusion

The graph representation of a board design has proven to be a versatile tool that allows for relative easy computation of the false paths and the impedances between certain nets. It allows unreachable structures to be collapsed into reachable compound structures and decomposing the graph into its biconnected components greatly reduces the mapping complexity. The minimal guard set algorithm finds all possible combinations of vertices to guard. If these guard sets are computed, it can then be determined what the maximal voltage is that can be applied to a net.

Tests in a pre-defined library of tests can be mapped onto the structures in the DUT. These tests specify a so called mold, which determines what kind of structures can be covered by that test. If the mold matches the mapping procedure continues. Structures not covered by a tests are then handled by the generic test method. By utilizing the PCOLA-SOQ properties the effectiveness of a proposed test can be optimize and guaranteed, and superfluous tests are automatically removed.

Future work entails adding support for: (1) active component, (2) shorts testing, (3) automatically selecting the most stable test.

References

- [1] K. Hird, K. P. Parker, and B. Follis. Test coverage: what does it mean when a board test passes? In *Proceedings IEEE International Test Conference (ITC)*, pages 1066–1074, 2002. doi: 10.1109/TEST.2002.1041863.
- [2] A. W. Ley. Defect coverage of non-intrusive board tests (nbt): What does it mean when a non-intrusive board test passes? In *2009 International Test Conference*, pages 1–1, Nov 2009. doi: 10.1109/TEST.2009.5355828.
- [3] Ieee standard for test access port and boundary-scan architecture. *IEEE Std 1149.1-2013 (Revision of IEEE Std 1149.1-2001)*, pages 1–444, May 2013. doi: 10.1109/IEEESTD.2013.6515989.
- [4] R. Matheson. Second-generation pcb self-learn. *Computer-Aided Engineering Journal*, 4(5):209–212, October 1987. doi: 10.1049/cae:19870051.
- [5] IEEE Standard American National Standard Canadian Standard Graphic Symbols for Electrical and Electronics Diagrams (Including Reference Designation Letters). *IEEE Std 315-1975 (Reaffirmed 1993)*, pages 241–244, 1993. doi: 10.1109/IEEESTD.1993.93397.
- [6] F. Harary. *Graph Theory*. Addison-Wesley series in mathematics. Addison-Wesley Publishing Company, 1969.
- [7] Wenxue Du. On graph isomorphism problem. *arXiv preprint arXiv:1710.09526*, 2017.
- [8] Chung-Wen Ho, A. Ruehli, and P. Brennan. The Modified Nodal Approach to Network Analysis. *IEEE Transactions on Circuits and Systems*, 22(6):504–509, Jun 1975. doi: 10.1109/TCS.1975.1084079.
- [9] Victor E Kane. Process capability indices. *Journal of quality technology*, 18(1):41–52, 1986.