Graph Theory Approach for Automatic Test Board Parameter Extraction in Multi-site IC Testing

Abraham Steenhoek*, Praise O. Farayola*, Isaac Bruce*, Shravan K. Chaganti†, Abalhassan Sheikh†, Srivaths Ravi†, and Degang Chen*.

*Iowa State University, Ames, USA.
†Texas Instruments Inc.

***Abstract*— This paper describes a low-cost technique for extracting parameters of interest for test boards used in a multi-site automatic test equipment (ATE). This paper addresses the need for automated extraction of test board parameters necessary for site-to-site variation investigations in massive multi-site testing. While multi-site testing significantly reduces test cost and time by testing chips simultaneously, it is often plagued with variations from site to site which do not correspond to actual problems in the device under test (DUT). Troubleshooting involves inspection and parameter extraction from the interface between the DUT and ATE (test board). This is a complex, multi-layered PCB whose design and fabrication are crucial to maintaining high signal integrity and testing accuracy. In the proposed approach, physical elements and nets on the PCB are represented with a graph with nodes and edges. Graph traversal algorithms are then used to extract data about the connections between specific components on each test site. This approach automates the previously slow and manual process of generating the topology files necessary to extract board parameters. In addition to board verification, extracted board parameters correlated with volume test data from the same tester/probe card configuration help to determine root causes of site-to-site variations in multi-site test hardware. The proposed method is implemented on a multi-site test board. The results and discussion are presented.**

Keywords—Multi-site Testing(MST), Automatic Test Equipment (ATE), Graph Theory, Site to Site (s2s) Variations, Test Board.

I. Introduction

Multi-site testing reduces test costs and time by simultaneously testing integrated circuits (ICs). It can be used both at the wafer probe and final tests, hence its wide acceptance and use. The method reduces all test contributors, not only the capital cost of ATE, and has been proven to be significantly cheaper than low-cost sequential ATE [1], [2].

One of the demerits of the method is site-to-site (S2S) variations in test measurements. By this, we mean unwanted tester/probe card influence on test results resulting in test site dependency [3]. These variations may arise from differences in tester resource arrangements, crosstalk, dielectric loss, prober, or handler alignment, amongst other reasons. They become more pronounced in massive multi-site (a large number of test sites) due to the increased difficulty in routing resources equally to all test sites. If not carefully monitored, S2S variations can lead to wrongful labeling of good ICs as "faulty" and loosening of test specifications limits for passed ICs.

Various statistical analysis on multi-site volume test data has been successfully used to identify issue test sites (test sites with unacceptable S2S variations) [4], [5]. Another approach is to analyze hardware characteristics of the multi-site (MST) probe card, known as board parameters (BPs). Board parameters encompass any parameter related to the physical characteristic of the PCB, including RLC of components, coupling coefficients between traces, trace length & width, number of vias in a trace, and regions on the board that a trace intersects.

Fig. 1 shows the essential parts of a multi-site tester. The printed circuit board (PCB) is responsible for routing the testing procedures (ATE resources) to each test site. The test board contains pads, vias, layers, transmission lines, pin mappings, contactors, and other components that may affect the accuracy of test measurements [6]. Historically, extracting board parameters has been a difficult process because there are not many automated end-to-end solutions. As MST ATE designs increase in complexity, having any kind of manual component in the data extraction process becomes an exponentially increasing issue. Additionally, the existing automated solutions do not provide information about BPs at a useful level of context and detail (discussed further in Section II).

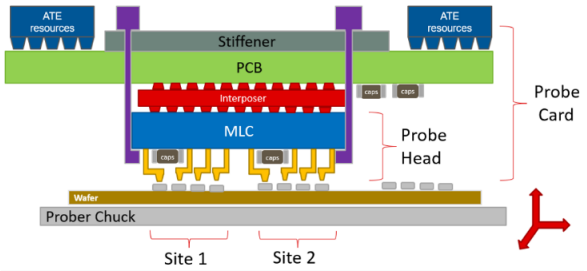

Fig. 1. Essential Parts of a Multi-site Test System

In this paper, we propose a novel method to extract board parameters from existing MST ATE designs using graph theory. We represent the physical elements on the PCB and their connects in a graph with nodes and edges, and use graph traversal algorithms to extract data about the connections between specific components on each site of the MST. This approach allows engineers to automate the collection of board parameters from dense and complex MST designs, and the graph representation provides a rich contextual insight into the connections between components on each site of the MST. In addition, test engineering teams can use this approach to

perform test board validation of their probe cards, saving development time and costs.

This paper is organized as follows: Section II provides motivation for the paper and problem statement. Section III introduces our proposed method and implementation. Section IV provides experimental results for the implementation in Section III. Section V discusses inherent limitations of the method and visualization tool for extracted parameters with a multi-site perspective. Section VI concludes the paper.

II. Motivation and Problem Statement

The test board (interface between the DUT and ATE) is becoming more critical to accurate testing than ever before. Analog and mixed-signal circuits (AMS), emerging high-speed devices, and other complex circuits put severe performance requirements on the ATE, especially the test board design and fabrication [6]. A high integrity test board requires much more complex and high-level design techniques than would have been required years ago. The problem compounds for multi-site test hardware.

Test boards go through a rigorous process to ensure that they are designed correctly. In [6], steps to a successful test board project are discussed, with each stage's susceptibility to errors analyzed. Compared to PCBs used in other fields, test boards in multi-site hardware have features of high thickness, high aspect ratio, high accuracy request, and high signal performance request [7]. They can become faulty because of the failure of components on the board while in use. This creates a need to find techniques to test the board and extract board parameters either before they are employed for IC testing or at regular intervals during use to ensure that the boards remain good [8].

The current methods in the market for testing PCBs are very expensive, time-consuming, and have limitations in their capability by the tester instrumentation. In [8], there is a need for a component library which can be time-consuming hence the need for other extracting methods and tools.

The concept of Xnets is important to understand when evaluating different BP extraction tools. Once a PCB layout is created in Allegro, the software makes a list of nets and Xnets on the PCB. A net is one connection node on a schematic. An Xnet is the result of an automatic process performed by Allegro, where all of the nets that contain only discrete components are lumped together into one summed Xnet. For example, the thick red section highlighted in Fig. 2 is a net, while the net touching the positive side of R1 and C2, and the net touching the negative side of C1, R1, and C2, is one Xnet. If we are interested in the length of the path from F1 to C2, Xnets can be leveraged in BP extraction.

Xnets provide valuable information for S2S variation analysis. It is crucial to have a feature that enables the study of these Xnets. Additionally, this feature/tool must be easy to use and be adaptable to various board parameters of interest. While selecting a board parameter of interest is easy, the tools available to extract this information from probe card PCB designs leave much to be desired. We briefly review some of existing tools in our PCB design suite, Cadence PCB Design Suite, including Allegro (for PCB schematic/layout editing) and SigXplorer (for modeling /simulations of aspects of the PCB).

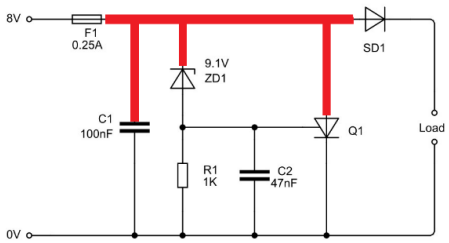

Fig. 2. Quick reports can only extract the etch length of the entire trace, but what if we want the etch of F1 to C2?

The first evaluated option was Allegro's reporting tools. Allegro has a quick-report feature that provides a report of the total trace length of all of the segments in a net as a grouped unit, as well as the length of individual segments in a single net. While this gives good fine-grained info about nets, this information is difficult to use when trying to determine the trace segments that connect two components. Allegro also has another reporting tool (different from the quick-report feature) that does include the etch length between all "pin pairs" on a net, where a pin pair is defined as a unique pair of passive components that share the same net. Although the Allegro built-in tools provide sufficient degrees of automation, they are not as robust as they are not built to support Xnets, which means that information about connections between components across separate nets cannot be obtained.

The second evaluated tool was SigXplorer. While SigXplorer is able to open Xnet files in its editor window as shown in Fig. 3, it has limited capabilities for automation. Although SigXplorer has the ability to extract the trace length BPs for Xnets, its lack of ability to automate makes using it a tedious manual effort, especially in large MST designs, where there the number of Xnets increase exponentially with complexity.

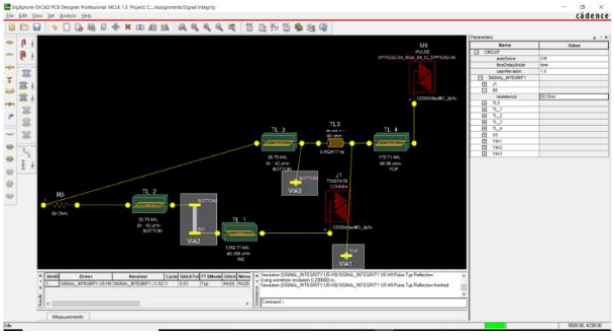

Fig. 3. Example Topology File opened in SigXplorer

The inability to find a tool that provided both the ability to analyze Xnets, as well as a usable automated extraction process, is the motivation for the tool presented in this paper. By representing Xnets as node graphs in software and by incorporating a scripted workflow that integrates with the Allegro PCB designer, we can analyze Xnets, as well as

automate the process in a way that is native to existing PCB design tools.

When determining the design of the tool, it is practical to think of how the Xnet data can be extracted. The Allegro tool stores information about every Xnet on the PCB in a topology file. A topology file, just like its name indicates, shows the "topology" of the PCB. Allegro's version of the topology file breaks every physical element on a net or a Xnet into nodes and displays the connections between those physical elements in the form of a nodal graph. The difference between a topology file and something like a schematic is that the topology file breaks down each trace into straight-line segments and shows where vias are used in the trace path.

Topology files can provide a lot of context about the layout of our multi-site tester. However, even if we assume that the expense for collecting topology files is covered, the data in each topology file still must be extracted and analyzed. Because SigXplorer (Cadence's tool for analyzing topology files) does not provide an automated way to extract data from topology files, we turn to a custom solution where we parse the files and extract data ourselves. Engineering a new method of data collection comes with a set of challenges and criteria. First, the potentially high number of sites indicates there will be many topology files; we must also be able to automate this topology file generation so that it is both practical and repeatable. Second, this method of data extraction must be flexible in terms of what components we want to measure. Finally, it is desirable for this data collection method to be able to calculate more than just the etch length, since the information stored in a topology file can possibly yield much more than that.

In this paper, we extract board parameters from topology files by analyzing each file as a set of nodes. By representing a section of a PCB as a node graph, we have a very flexible dataset with automated methods of data extraction thanks to graph theory. With tools in graph theory such as traversal algorithms and path-finding algorithms, we can automate the process of collecting observations that previously would need to be done visually by a test/design engineer, or possible even a team. For example, engineers can define board parameters or associations of interest between components and use a traversal algorithm to collect data about this board parameter from every site on their multi-site tester.

In this paper, we build a tool that applies this concept to extract the etch length between any two components in a site. We call this tool the Node Graph Etch Length Extractor (NGELE), or simply Etch Length Extractor (ELE) for short. While the tool developed does extract the etch length between any two components in a topology file, the key novelty is in its *node graph approach*; the ELE portion is necessary to have a board parameter as a test subject to prove the concept.

III. Proposed Method and Implementation

a) Proposed Method

This section is discussed in a two-step process: First, topology file generation, and second, topology file analysis.

One crucial detail to consider is the process of how topology files are acquired in the first place. Although the engineer may know the nets/Xnets of interest, these nets must be available in a file format that can be parsed and analyzed with software. Cadence Allegro stores this information in topology files. Obtaining topology files was previously an arduous manual process but was resolved with the implementation of a software topology file generator tool.

The goal for topology generation is to design a tool that takes a list of net/Xnet names as an input and gives a corresponding list of topology files as an output. A natural option for automating file generation is scripting. Allegro provides a way to automate point-and-click tasks with its custom SKILL language. SKILL was a strong option, due to its native integration into the Allegro editor, relatively readable code style, and specific help and recommendation from Cadence Support. The list below shows the tasks that are required to generate a topology file:

- Open the Probe Card Layout project in Allegro
- Open the Constraint Manager (CM)
- Select the desired net or Xnet to generate a topology file in the CM's Netlist Viewer
- Right click on the desired file and click "Open in SigXplorer"
- In SigXplorer, click File->Save As ->[myfilename].top

The topology file generator (TFG) tool, implementing with a SKILL script, automates the process above. The operation of this script is summarized in Fig. 4.

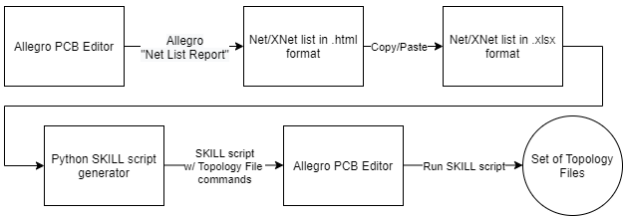

Fig. 4. System Diagram showing the flow of operation for the Python SKILL script

Once the topology files have been obtained, an understanding of the internal data structure must be understood to properly extract relevant board parameters. This helps develop a method that is not only useful for etch length, but also any number of board parameters that relate to the metadata inside of a topology file.

The goal for topology analysis part of this paper is to develop a tool that allows users to automate the extraction of the etch length between two components from a given topology file. Fig. 5 shows a simple diagram of the desired function.

Fig. 6 shows how components in an Xnet, along with their logical connections, are shown in the respective topology file the Xnet information is stored in. This topology file holds this data in a parse-able test format.

Fig. 5. Diagram of the Node Graph Etch Length Extractor (ELE)

Seeing the visual appearance of a topology file brings an interesting correlation to a graph with nodes and edges. Representing a topology file as a graph opens the door for classic graph traversal methods (DFS/BFS [8], Dijkstra's algorithm [4], etc.), which provides immense flexibility in extracting data that is related to sets of entities, or paths between linked entities. In theory, a graph representation of a topology file makes the extraction of the following information (and more) possible to entirely automate with software:

- Etch length between two (or more) components
- Number of vias between two (or more) components
- Number of turns in a trace connecting two components

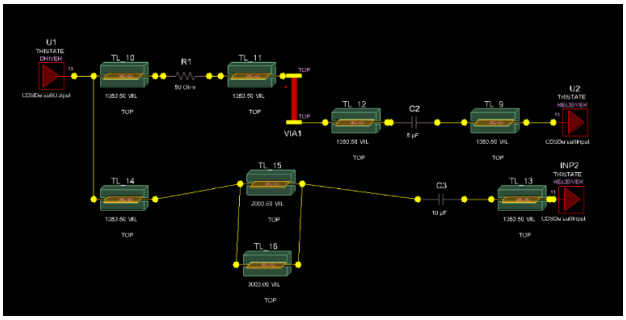

Fig. 6. Contents of example topology file: ex1.top

b) Implementation

Each topology contains all of the metadata for each component the Xnet, as well as that component's connections to other components, in a plaintext engineered by Cadence. This makes analyzing the contents much easier to do with a visual inspection, and makes extraction much easier through software text parsing. The information in each topology file is entirely sufficient to properly translate that topology file into a node graph.

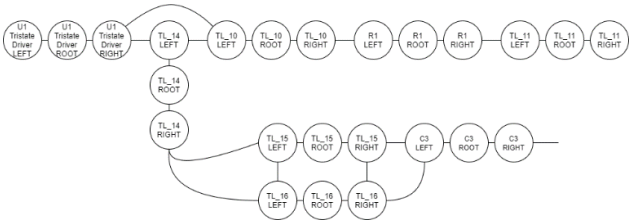

Fig. 7. Applying the L-R-Root conversion to the starting concept

For easier reference, the topology file in Fig. 6 is referred to as ex1.top. While all the necessary information to create a nodal graph from ex1.top is available in its topology file, the translation from a topology file to a node graph must be done carefully to ensure the accuracy of representation.

Fig. 7 shows a conversion from the ex1.top file to a nodal graph. It is important to notice the LEFT, ROOT, and RIGHT nodes in the design. A naïve approach may suggest representing each component in ex1.top as a single node, but does not accurately represent the component, because each component has a left and right side, each with different connections. The LEFT and RIGHT nodes are necessary for an accurate representation of the connections of traces to each side of a resistor/capacitor/via/etc. The ROOT node is introduced in between the LEFT/RIGHT nodes as an element to help with the software implementation.

Developing a proper nodal graph representation provides a flexible and contextual dataset, where powerful graph traversal algorithms can be leveraged to automate data extraction in software. In the case of this paper, determining the etch length between two components in a topology file is a perfect candidate for a path-finding algorithm. Plotting the etch length between two components involves determining the node path(s) from the start node to the end node and then summing up the etch length from all of the trace components in a node path. The algorithm of choice for finding the node path was the Depth-First Search Algorithm (DFS) [8] because it provides the most straightforward implementation to find *all possible paths* from the start to the end node. Visual inspection of our multi-site tester design showed multiple instances where two components were connected by two unique trace paths (Fig. 8), proving the value of the ability to plot multiple paths.

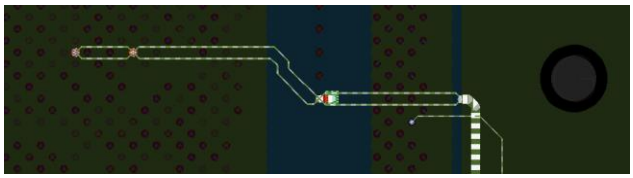

Fig. 8. ELE Tool operation from the user perspective

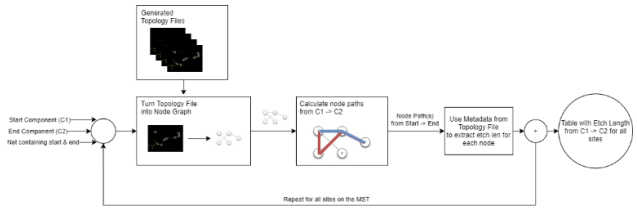

Fig. 9. ELE Tool operation from the user perspective

Fig. 9 shows the operation of the ELE tool from the user perspective. With the operation of the TFG and ELE tools described, we provide an example of how a test engineer might use them in conjunction.

- Determine a path of interest: e.g., Etch length from R1 -> C1 for all sites
- Use the TFG to generate topology files for the path from R1 -> C1 for all sites on the probe card
- Use the ELE to generate a table of the etch length from R1 -> C1 for all sites, using the TFG-generated files

IV. Experimental Results

Fig. 10 shows the output from running the ELE tool on one of the massive multi-site testers from Texas Instruments. The output in Fig. 10 shows that the tool has calculated the trace length from C1 to C2 for all sites on the MST. If there happen to be multiple paths from C1 to C2 on a site, the tool includes the information for both paths. The trace lengths for instances where there are multiple paths from C1->C2 are also calculated independently.

Functionally, the ELE tool can provide the same information as the Allegro resources for nets. However, for Xnets, Allegro does not have a satisfactory solution. Since the ELE tool does add detailed and contextual analysis of Xnets, it can be considered as a strong supplement to Allegro for BP analysis.

This comparison both demonstrates Allegro's capabilities for those who need a quicker or more generic solution than the ELE tool, and highlights the benefits of using a self-developed tool for specific applications where specific board parameters that are out of reach for the Allegro reporting tools are needed.

	A	B	C
1	C1.C2.DIST.1	C1.C2.DIST.2	SITE
[redacted]	4409.941		[redacted]
	7755.13		
	4608.838	4608.496	
	5736.229		
	4608.305		
	3617.021	3617.124	
	4647.202	4647.305	
	5372.46	5372.118	

Fig. 10. Section of output from running ELE tool with the command shown in Figure 15

V. Discussion

In this paper, the proposed method was proven using the etch length BP. However, the tool can be adapted to analyze other kinds of information with some simple software additions. Because the implementation of this method relies primarily on the concept of representing the topology file as a graph, it is flexible to extracting any kind of information that is available in the source (topology) files from where the graphs are generated. The topology file may include any parameters relevant to the physical properties of the PCB, for example, the number of vias between two components, trace impedance between two components, and the number of trace segments used to route two components.

The idea of turning a topology file into a node graph can be taken a step further. The entire PCB of a MST is modeled by sets of separate topology files. It is possible that these topology files can be joined together by the ELE tool by turning each topology file into a node graph, and then stitching the nodal graphs together. Fig. 11 shows a limitation of the current ELE tool, where board parameters for relations between {R1, R2} to {R3,R4} cannot be extracted because Xnets stop at an active component. Combining the node graphs for the topology files open the capability to analyze BPs across an entire site on the MST, BPs from site to site, or even BPs across *the entire PCB*. Node graph representation can be used to gain a much deeper contextual understanding of the physical connections between components, giving the engineers a rich tool for performing extensive *test board* validation of their circuit designs.

Another limitation of this approach is its inability to extract parameters that may require simulation engines in the PCB CAD software. This includes parameters involving n-port models (S, Z, ABCD parameters), signal integrity parameters, and dynamic voltages and currents. This is because the graph approach extracts information relating mainly to the layout of the MST PCB.

Visualization plays a vital role in extracted data analysis by placing the extracted data in terms of visual context. The right visualization tool will give an accurate insight into data and find useful information that may have otherwise been missed [9], [10].

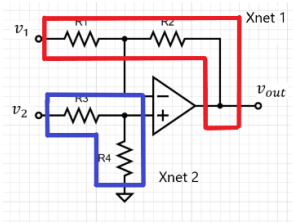

Fig. 11: XNets cannot be used to measure board parameters that span across active components

Mainly two-dimensional plots are used to visualize data set containing two variables. Fig. 12A is an example of the extracted trace length between two pins plotted against site for a 49-test site multi-site tester. While this method is simple, popular, and easy to draw, it hides some data structures that may be important to the test engineer and stringent to the understanding of S2S variations in multi-site testing.

The 3-D representation of data gives more knowledge to the user as it is able to express information in a three-dimensional space. In Fig. 12B, we present a modified 3D bar chart used to visualize the same dataset presented in Fig. 12A. The length values are not included for privacy reasons. One important addition is the test site layout perspective factored into the bar charts. We observe an interesting well curve not evident in Fig. 12A. This is important and might lead to correlation with the mean/median of test measurements from each test site.

VI. Conclusion

In this paper, two tools were introduced to assist in the extraction of test board parameters. The Topology File Generator (TFG) automates the previously manual process of generating the topology files necessary to extract board parameters, now making it a trivial and inexpensive effort. The ELE tool provides a method of automating the extraction of data from topology files by representing each group of components in a topology file as a node graph. The node graph approach allows future developers to attach new software

functions to the existing NGELE tool to extract new board parameters relating to topology file data with a relatively low software development effort.

The graph theory approach introduces a new method of multi-site tester analysis to the automated testing community. The contextuality and amenability of graphs with nodes and edges increase the data available in site-to-site analysis by adding information about the relationships of circuit components used to implement each site and even components across multiple sites. Bringing S2S analysis down to the hardware implementation of the MST not only helps identify root causes for future MSTs, but also perform test board validations at the pre-fabrication stage.

The goal is to find a correlation between the extracted board parameters and test volume data. These correlations can be used to narrow down trouble areas on the PCB to search for flaws in the hardware design of the MST. This goal is mentioned to reinforce the reasoning behind the board parameter extraction efforts in this project. However, the process of finding correlations between Board Parameters and Die Parameters is outside the scope of this paper.

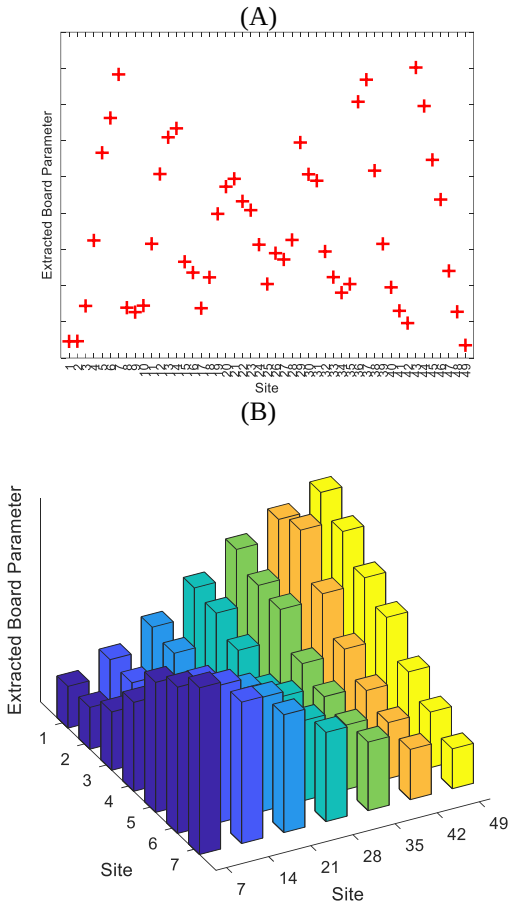

Fig. 12 2D and 3D representation of an extracted trace length between two nodes for a 49-test site board

ACKNOWLEDGMENT

This work was supported by Texas Instruments and the Semiconductor Research Corporation.

REFERENCES

[1] J. Rivoir, "Parallel test reduces cost of test more effectively than just a cheap tester," in *IEEE/CPMT/SEMI 29th International Electronics Manufacturing Technology Symposium (IEEE Cat. No.04CH37585)*, Jul. 2004, pp. 263–272. doi: 10.1109/IEMT.2004.1321674.

[2] Lew Boon Kian, "Test cost saving and challenges in the implementation of /spl times/6 and /spl times/8 parallel testing on freescale 16-bit HCS12 microcontroller product family," in *Third IEEE International Workshop on Electronic Design, Test and Applications (DELTA'06)*, Jan. 2006, p. 7 pp. – 82. doi: 10.1109/DELTA.2006.85.

[3] T. Lehner, A. Kuhr, M. Wahl, and R. Brück, "Site dependencies in a multi-site testing environment," in *2014 19th IEEE European Test Symposium (ETS)*, May 2014, pp. 1–6. doi: 10.1109/ETS.2014.6847808.

[4] A. Al-Obaidi, "Reducing integrated circuit test cost through improvements in multi-site testing and built-in self-test," *Grad. Theses Diss.*, Jan. 2020, doi: https://doi.org/10.31274/etd-20200902-3.

[5] P. O. Farayola, S. K. Chaganti, A. O. Obaidi, A. Sheikh, S. Ravi, and D. Chen, "Quantile – Quantile Fitting Approach to Detect Site to Site Variations in Massive Multi-site Testing," in *2020 IEEE 38th VLSI Test Symposium (VTS)*, Apr. 2020, pp. 1–6. doi: 10.1109/VTS48691.2020.9107616.

[6] D. E. McFeely, "The process and challenges of a high-speed DUT board project," in *Proceedings. International Test Conference*, Oct. 2002, pp. 565–573. doi: 10.1109/TEST.2002.1041807.

[7] J. Liu, F. Pan, and D. Zhou, "The design of multilayer IC test board based on ATE," in *2017 18th International Conference on Electronic Packaging Technology (ICEPT)*, Aug. 2017, pp. 1552–1556. doi: 10.1109/ICEPT.2017.8046731.

[8] S. Kannan and B. C. Kim, "Automatic diagnostic tool for Analog-Mixed Signal and RF load boards," in *2009 International Test Conference*, Nov. 2009, pp. 1–1. doi: 10.1109/TEST.2009.5355836.

[9] R. S. Raghav, S. Pothula, T. Vengattaraman, and D. Ponnurangam, "A survey of data visualization tools for analyzing large volume of data in big data platform," in *2016 International Conference on Communication and Electronics Systems (ICCES)*, Oct. 2016, pp. 1–6. doi: 10.1109/CESYS.2016.7889976.

[10] S. M. Ali, N. Gupta, G. K. Nayak, and R. K. Lenka, "Big data visualization: Tools and challenges," in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, Dec. 2016, pp. 656–660. doi: 10.1109/IC3I.2016.7918044.