ITERATIVE DIAGNOSIS TO IMPROVE DIAGNOSTIC RESOLUTION

ANDREW CHUAH HOOI LEONG

UNIVERSITI TEKNOLOGI MALAYSIA

ITERATIVE DIAGNOSIS TO IMPROVE DIAGNOSTIC RESOLUTION

ANDREW CHUAH HOOI LEONG

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Electrical - Computer and Microelectronic System)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

JANUARY 2013

Especially to Lai Lin, Jayna, Jalynn and my parents.
Thank you for your unending encouragement and support.

ACKNOWLEDGEMENT

I would like to take this opportunity to express my deepest gratitude to my project supervisor, Dr. Shaikh Nasir bin Shaikh Husin for his encouragement, guidance and sharing of knowledge throughout the process of completing this project.

I would like to extend my appreciation to Intel Microelectronics (M) Sdn. Bhd., for funding my studies. I would like to thank my mentor in the field of diagnosis, Dr. Srikanth Venkataraman for his initial ideas and thoughts on the project direction. My thanks also go out to my colleagues, Dr. Enamul Amyeen and Carlston Lim, who were always open and willing to answer my questions on fault simulation and diagnosis tools.

Last but not least, I would like to thank my parents, Mr. Chuah Tong Ik and Dr. Seow Siew Hua, who helped immensely by proofreading this report and suggesting improvements. My deepest gratitude also to my wife and daughters, for their selfless support, encouragement and love.

ABSTRACT

The area of research is the study of iterative diagnosis. Diagnosis to find faults in semiconductor devices is a well researched field, with most logic diagnosis efforts using the inject-and-evaluate algorithm. However, most diagnosis tools are unable to resolve faults to a single gate/device. Because of this, fault isolation (FI) engineers are forced to use probing techniques such as IREM logic state imaging (LSI) in order to further isolate the fault to the gate/device level before performing failure analysis. The current method of selecting probe sites is simply to take the list of fault candidates and probe them sequentially or by determining the optimal probe order through manual analysis of the circuit cone. However, in cases where a large list of fault candidates are returned by the diagnosis tool, it is difficult to manually analyze the fault cone as it is too large and complex. This work implements a basic algorithm which allows the diagnosis tool to recommend probe candidates, read in the result of the probe, and continue this cycle iteratively until the fault is fully isolated to a single gate/device. The algorithm is based on a binary search, and shows that a 5-6X reduction in the amount of probing needed can be achieved if the diagnosis tool is used iteratively in the fault isolation flow.

ABSTRAK

Bidang penyelidikan yang dikaji adalah diagnosis iteratif (iterative diagnosis). Diagnosis untuk mencari kecacatan dalam alat semikonduktor merupakan suatu bidang yang banyak dikaji. Kebanyakan usaha diagnosis lojik menggunakan algoritma “inject-and-evaluate”. Walau bagaimanapun kebanyakan alat diagnostik tidak dapat menyelesaikan kesalahan mengenai alat/pintu asas (single gate/device). Oleh itu, jurutera pencarian kecacatan (fault isolation) terpaksa menggunakan teknik penyelidikan seperti “IREM logic state imaging” (LSI) untuk mengasingkan lagi kecacatan terhadap paras alat/pintu sebelum menjalankan analisis kegagalan. Kaedah sekarang yang digunakan untuk memilih tapak kajian (probe sites) ialah dengan menggunakan senarai tapak-tapak kesalahan (fault candidates) dan mengkajinya secara satu demi satu, atau dengan menentukan susunan tapak kajian optimis (optimal probe order) melalui menganalisis kun litar (circuit cone). Walau bagaimanapun dalam kes dimana banyak tapak kesalahan dikesan oleh alat diagnostik, amatlah sukar untuk meneliti kun kesalahan (fault cone) kerana ianya terlalu rumit dan besar. Kajian ini melaksanakan suatu algoritma asas yang digunakan oleh alat diagnostik untuk mencadangkan tapak yang perlu dikaji. Setiap keputusan kajian kemudian dihantar semula kepada alat diagnostik dan pusingan ini diteruskan sehingga kecacatan diasingkan ke hanya satu pintu/alat (single gate/device). Algoritma ini berasaskan pencarian binari dan telah menunjukkan bahawa kekurangan kerja sebanyak 5-6 kali boleh dicapai sekiranya alat diagnostik digunakan secara iteratif dalam proses pengasingan kecacatan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ATE	-	Automated Test Equipment
CAD	-	Computer Aided Design
CUT	-	Circuit Under Test
DTPG	-	Diagnostic Test Pattern Generation
DUT	-	Device Under Test
FA	-	Failure Analysis
FI	-	Fault Isolation
FM	-	Fidduccia-Mattheyses algorithm
IREM	-	Infrared Emission Microscopy
LADA	-	Laser Assisted Device Alteration
LSI	-	Logic State Imaging
LVI	-	Laser Voltage Imaging
LVP	-	Laser Voltage Probing
LVS	-	Layout vs. Schematic verification
MHEC	-	Modified Hyperedge Coarsening algorithm
MLFM	-	Multilevel Fiduccia-Mattheyses algorithm
MOSFET	-	Metal-Oxide-Semiconductor Field Effect Transistor
NMOS	-	N-type Metal-Oxide-Semiconductor Transistor
NP	-	Nondeterministic Polynomial Time
PEM	-	Photon Emission Microscopy
PFA	-	Physical Failure Analysis
PICA	-	Picosecond Imaging Circuit Analysis
PMOS	-	P-type Metal-Oxide-Semiconductor Transistor

LIST OF ABBREVIATIONS

RAM	-	Random Access Memory
ROI	-	Region Of Interest
SA	-	Stuck At
SEM	-	Scanning Electron Microscopy
SRAM	-	Static Random Access Memory
TCL	-	Tool Control Language
TEM	-	Transmission Electron Microscopy
TRE	-	Time Resolved Emission
XML	-	Extended Markup Language
XPath	-	XML Path Language

CHAPTER 1

INTRODUCTION

1.1 Background

As transistors continue to shrink, fault isolation needs to improve in resolution in order to have accurate failure analysis [1]. As a result, more probing is done in each subsequent technology generation, resulting in greater throughput time for the fault isolation (FI) and failure analysis (FA) process.

During the integrated circuit (IC) manufacturing cycle, manufacturing tests are generated that either functionally or structurally (using scan or other design for test (DFT) mechanisms) test the device under test (DUT). These tests are applied using a tester (sometimes called Automated Test Equipment, or ATE). The tests are simply vectors of 1's and 0's and the outputs of the DUT are strobed for outputs that match the "golden" output that was generated during test generation [2].

During this process, certain parts will be found to be defective and will be sent for FI, in order to narrow down the physical location of the defect, and then FA using scanning electronic microscopy (SEM) systems can be performed in order to determine the exact defect mechanism. Once the defect mechanism is determined, the fabrication process can then be examined in order to understand and fix the root cause in the fabrication process, equipment or methodology causing the defect. This cycle is the process used by all semiconductor manufacturing companies to improve yield, which ultimately lowers the cost of the product.

Figure 1 shows the feedback loop between manufacturing, yield analysis and failure analysis. The time taken to complete a cycle of the loop is of utmost importance, as it determines how much yield learning can be done in a period of time. If cycle time reduces, yield learning is faster, and results in an equivalent drop in the cost of production.

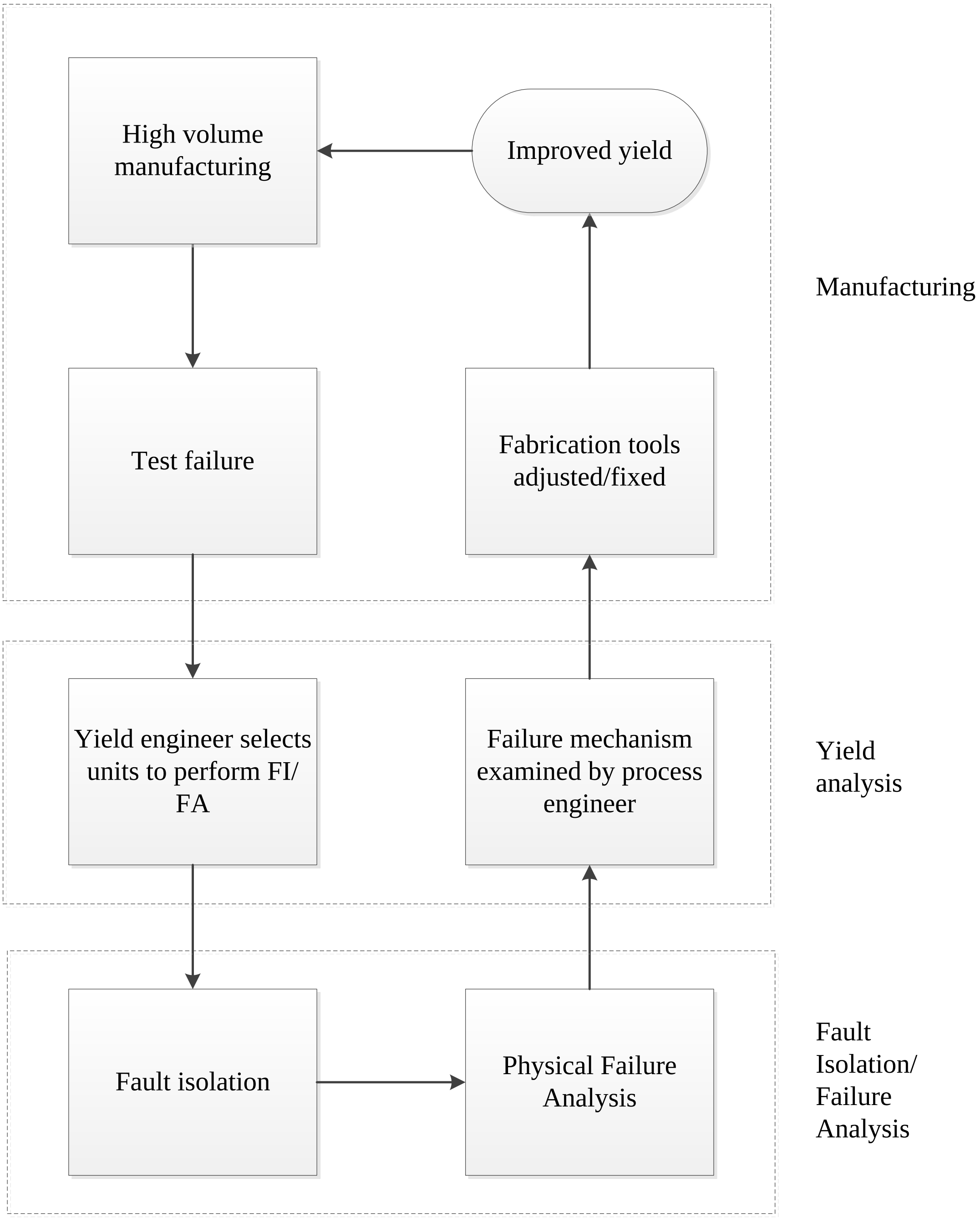
High volume manufacturing
Improved yield
Manufacturing
Test failure
Fabrication tools adjusted/fixed
Yield engineer selects units to perform FI/FA
Failure mechanism examined by process engineer
Yield analysis
Fault isolation
Physical Failure Analysis
Fault Isolation/ Failure Analysis

Figure 1.1 Manufacturing-yield analysis-FI/FA feedback loop

This study will focus on the Fault Isolation step in Figure 1.1. We will introduce the current workflow in fault isolation, and suggest a workflow that will improve the throughput time of fault isolation, and thereby speed up the overall time to complete the manufacturing-yield analysis-FI/FA loop.

1.2 Motivation and Problem Statement

As the number of gates on a single IC continue to grow, the effort of finding defects on silicon also increases. As a result, failure analysis teams continue to rely on optical probing tools such as photoemission microscopy (PEM) in order to reduce the region-of-interest (ROI) before physical failure analysis (PFA) can be carried out. This activity is called Fault Isolation (FI), as it attempts to isolate the fault into a small region for imaging. Figure 1.2 illustrates that the ability of several techniques in slowly reducing the ROI. These activities are often done sequentially, in order to reduce the ROI. For example, analyzing scan test fails allow us to narrow the ROI to a block in design. The scan fails are then entered into as an input to the diagnosis tool, which produces a list of diagnosis candidates (further narrowing the ROI to a list of gates/wires). These candidates can then be probed by various tools, which narrow down the list to a single gate/wire, then further on to part of a wire or transistor.

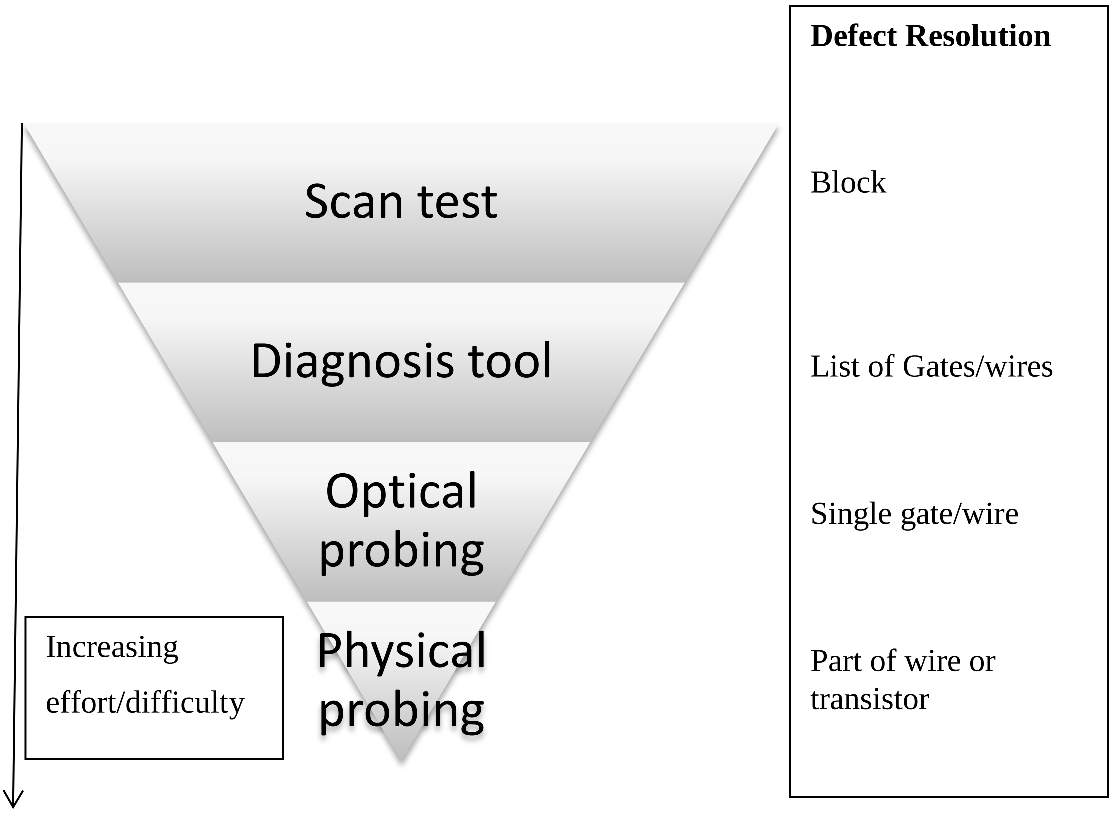

Figure 1.2 Resolution vs. effort in fault isolation techniques.

Due to the increase in effort between running the diagnosis tool and performing optical probing, it is critical that the diagnosis tool returns the minimal number of defect candidates. There has been much effort by researchers to improve the resolution of diagnosis, and there have been multiple improvements in the past years. These improvements include layout aware diagnosis, diagnostic test pattern generation, intra-cell diagnosis and additional fault types. These efforts are further elaborated in section 2.1.3. Despite these improvements, there are still many instances whereby the diagnosis tool is unable to return a small number of candidates. When this happens, the optical probing activity effort increases substantially, as the list of diagnosis candidates that have to be probed is very large. Therefore, there is a need to improve the workflow of optical probing activity by reducing the probing time for cases with unsatisfactory diagnosis tool output.

Figure 1.3 shows a typical FI/FA flow. This diagram sheds further detail on the optical and physical probing activities. Both activities require pre-work to find the correct location to probe, and then actually using the tool/equipment to probe the area. In this study, we will limit ourselves to the diagnosis tool and optical probing activities, as shaded in Figure 1.3. We will also only restrict our discussion on optical probing to the photoemission microscopy (PEM) and infrared microscopy (IREM) tools.

In the current shaded flow in Figure 1.3, the steps of the flow are run sequentially (diagnosis, then optical probing). This is the main problem in the current flow. It assumes that after diagnosis is run, the diagnosis tool can no longer contribute to FI. As a result, the probing activity takes a long time if the diagnosis tool returns a large list of possible candidates, as they all have to be probed. Probing can take from hours to days depending on how many signals have to be observed.

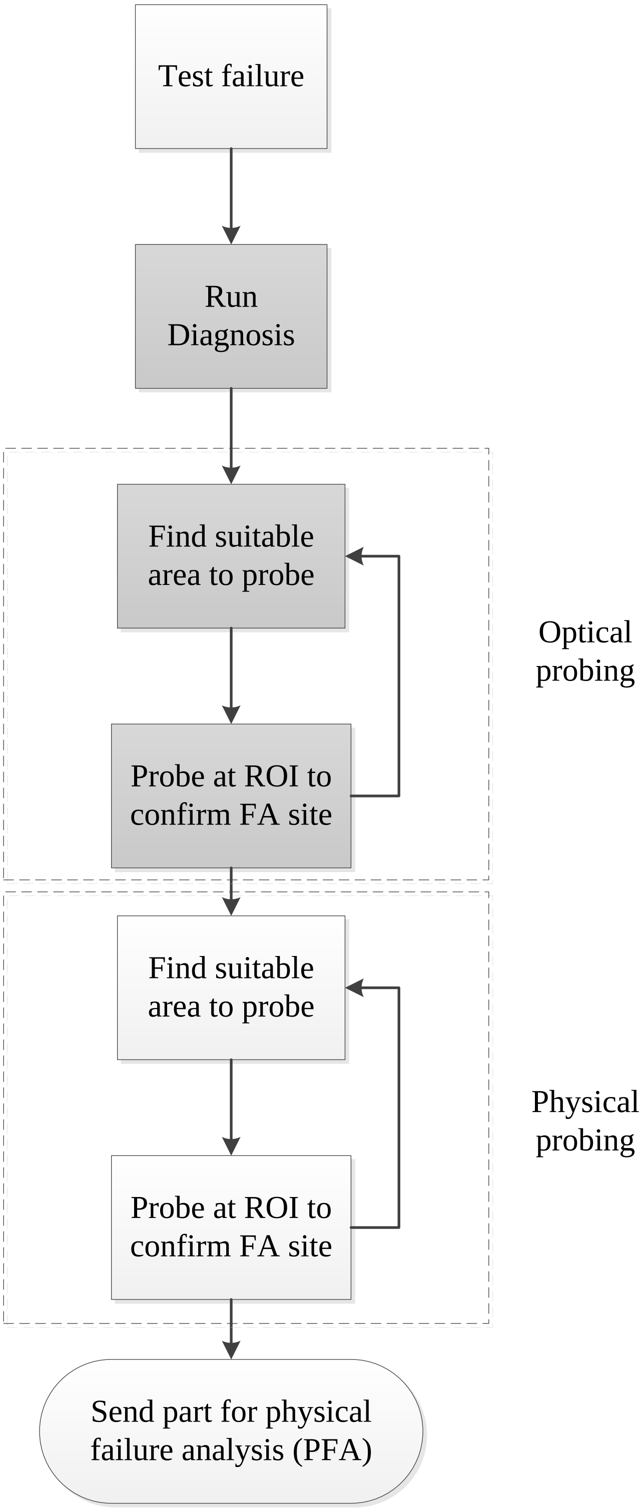
Test failure
Run Diagnosis
Find suitable area to probe
Probe at ROI to confirm FA site
Optical probing
Find suitable area to probe
Probe at ROI to confirm FA site
Physical probing
Send part for physical failure analysis (PFA)

Figure 1.3 Typical fault isolation flow.

Besides long probing time, another challenge for the FI engineers is that IREM logic state imaging (LSI) probing (detailed further in section 2.2.1) can only be performed on certain devices. The conditions where a sufficiently bright emission can be detected by the IREM tool differ for each fabrication process, as well as the IREM tool detection sensitivity. Inverters have generally have high enough subthreshold leakage to appear clearly on IREM results. However, other gates may have stacked transistors, which will not produce emissions. Therefore, not every signal that is returned by the diagnosis tool can be physically probed by the IREM. FI engineers often have to trace forward or backward in the circuit to find a "probe point" that allow accurate probing and also will imply the logic value on the diagnosis signal. This information is already present in the diagnosis tool, but due to the sequential nature of the flow, the FI engineer has to manually determine the correct devices to probe.

1.3 Objectives

This study attempts to find and implement a basic algorithm which allows the diagnosis tool to recommend probe candidates, read in the result of the probe, and continue this cycle iteratively until the fault is fully isolated to a single gate. The aim will be to minimize the number of probe frames needed by introducing a feedback and feed forward mechanism between probing and diagnosis applications.

In the current flow (Figure 1.3), the diagnosis tool merely returns the list of probe candidates. This list could be large, in cases where the diagnosis tool has not been able to narrow down the list of candidates sufficiently. In our new proposed flow (Figure 1.4), we extend the diagnosis tool to return a carefully chosen small list of candidates to probe first, if the list is too large. The probe results of the small list are then returned to the diagnosis tool. The tool then performs an incremental diagnosis run, taking as input probe results and producing another small list of candidates to probe. This iterative process continues until the list of candidates have been narrowed to one. The shaded portion of Figure 1.4 shows the process that has just been described.

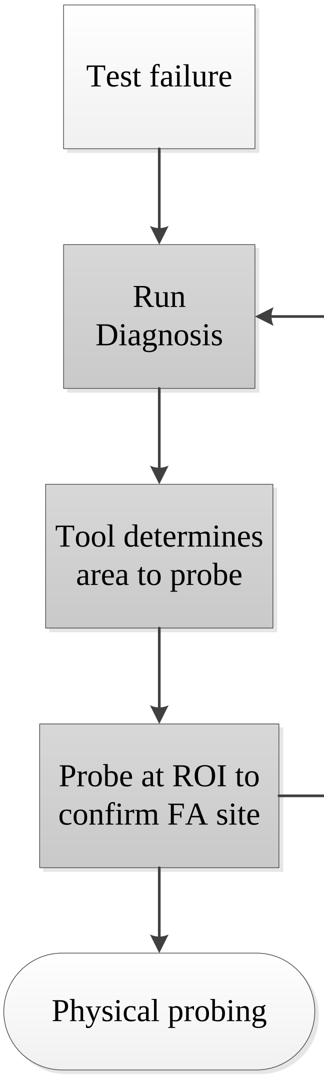

Figure 1.4 Iterative diagnosis workflow

There is no documentation of this iterative loop between the diagnosis tool and probing activity in our literature search. This study attempts to improve the fault isolation process with the iterative diagnosis workflow shown in Figure 1.4.

1.4 Scope of Work

In this study, we implemented the new workflow, as seen in Figure 1.4. The large majority of our focus is on determining the area to probe. However, in order to obtain results from the new workflow, we executed all the shaded steps in the new workflow. We also obtained results based on the new flow which show 5-6X improvement over the original flow. The results are however only based on a single testcase, with a single defect.

In order to determine the area to probe, the algorithm goes through the following steps: First, it parses the diagnosis report file from the diagnosis tool in order to identify the candidates and mismatches. (Further explanation of diagnosis candidates, mismatches and diagnosis algorithms are available in section 2.1.1). It then creates a graph from the diagnosis cone. Then, it prunes the graph and drops non-probeable gates. (Probe-able gates are explained in section 2.2.1). The graph is then weighted and bisected. (Graph bisection is discussed in section 2.3 and 2.4).

Although we have limited our scope to IREM LSI probing in this study, this flow can also be used in other observational, non-destructive probing techniques such as Laser Voltage Probing (LVP) and Time Resolved Emission (TRE) [3].

1.5 Report Outline

Chapter 1 has provided a brief introduction to the problem of fault isolation, and the different methods used to narrow down defect candidates to a small ROI (part of a device/interconnect) in order to perform defect imaging. We have also introduced the current flow between diagnosis and optical probing, and have proposed a new flow which iterates between them to improve efficiency.

Chapter 2 presents background information on semiconductor diagnosis, photoemission microscopy, binary search and graph partitioning. The workings and algorithms used in semiconductor diagnosis are explored in section 2.1.1, and the various efforts to improve diagnostic resolution in section 2.1.3. A brief introduction to PEM and IREM are given in section 2.2, and the concept of Logic State Imaging (LSI) is explained in section 2.2.1. Binary search and graph partitioning are both tools used in our algorithm, and the algorithms commonly seen in literature are introduced in sections 2.3 and 2.4.

Chapter 3 presents the basic algorithm which determines the probe locations. The steps involved are extracting a circuit cone from the candidates list, converting the cone into a graph, pruning the graph, and bisecting it to determine the probe points, and deriving the probe frames. In Chapter 4, we discuss the detailed implementation of the algorithm. This includes the programming language, data structures and search algorithms that were used in implementation. We also present the experimental setup and test case used to generate the results on this report.

Chapter 5 presents and compares the results from 2 different constraints applied to the graph bisection algorithm. In Chapter 6, we discuss other learnings and analyze the results gleaned from the testcase. Chapter 7 presents the conclusion of the thesis, and recommendations for future work. This is followed by the bibliography.

REFERENCES

1. Vallett, D. P. Why Waste Time on Roadmaps When We Don’t Have Cars? *IEEE Transactions on Device and Materials Reliability*. 2007. 7(1): 5-10.
2. Bushnell, M. L. and Agrawal, V. D., *Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits*. New York, New York, USA: Springer. 2000.
3. Chin, J. M., Narang, V., Zhao, X., Tay, M. Y., Phoa, A., Ravikumar, V. and Tan, M. C. Fault isolation in semiconductor product, process, physical and package failure analysis: Importance and overview. *Microelectronics Reliability*. 2011. 51(9): 1440-1448.
4. Segura, J., Keshavarzi, A., Soden, J., and Hawkins, C. Parametric failures in CMOS ICs - a defect-based analysis. *Proceedings of International Test Conference*. October 7-10, 2002. Baltimore, Maryland, USA. IEEE. 2002. 90-99.
5. Moore, G. E. Cramming more components onto integrated circuits. *Proceedings of the IEEE.* 1998. 86(1): 82-85.
6. Ghani, T. Challenges and Innovations in Nano-CMOS Transistor Scaling. *2010 IEEE Workshop on Microelectronics and Electron Devices*. April 16, 2010. Boise, Idaho, USA. IEEE. 2009.
7. Kuehlmann, A., Cheng, D. I., Srinivasan, A., and LaPotin, D. P. Error diagnosis for transistor-level verification. *Proceedings of the 31st Design Automation Conf*. June 6-10, 1994. San Diego, California, USA: IEEE. 1994. 218-223.
8. Venkataraman, S. and Drummonds, S. B. Poirot: applications of a logic fault diagnosis tool. *IEEE Design & Test of Computers.* 2001. 18(1): 19-30.
9. Wang, L. T., Wu, C. W., and Wen, X. *VLSI test principles and architectures: design for testability*. San Francisco, California, USA: Morgan Kaufmann. 2006.

10. Pomeranz, I., and Reddy, S. M. On correction of multiple design errors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.* 1995. 14(2): 255-264.

11. Waicukauski, J. A., and Lindbloom, E. Failure diagnosis of structured VLSI. *IEEE Design & Test of Computers*. 1989. 6(4): 49-60.

12. Bartenstein, T., Heaberlin, D., Huisman, L., and Sliwinski, D. Diagnosing combinational logic designs using the single location at-a-time (SLAT) paradigm. *Proceedings of International Test Conference.* October 30-November 1, 2001. Baltimore, Maryland, USA: IEEE. 2001. 287-296.

13. Rousset, A., Bosio, A., Girard, P., Landrault, C., Pravossoudovitch, S., and Virazel, A. Improving Diagnosis Resolution without Physical Information. *4th IEEE International Symposium on Electronic Design, Test and Applications*. January 23-25, 2008. Hong Kong: IEEE. 2008. 210-215.

14. Pomeranz, I., and Reddy, S. M. A Same/Different Fault Dictionary: An Extended Pass/Fail Fault Dictionary with Improved Diagnostic Resolution. *Design, Automation and Test in Europe, DATE'08*. March 10-14, 2008. Munich, Germany: IEEE. 2008. 1474-1479.

15. Yamazaki, I., Yamanaka, H., Ikeda, T., Takakura, M., and Sato, Y. An approach to improve the resolution of defect-based diagnosis. *Proceedings of the 10th Asian Test Symposium*. November 19-21, 2001. Kyoto, Japan: IEEE. 2001. 123-128.

16. Stanojevic, Z., Balachandran, H., Walker, D. M. H., Lakhani, F., and Jandhyala, S. Defect localization using physical design and electrical test information. *Advanced Semiconductor Manufacturing Conference and Workshop (ASMC).* September 12-14, 2000. Boston, Massachusetts, USA: IEEE. 2000. 108-115.

17. Ravikumar, C. P., Sharma, M., and Patney, R. K. Improving the diagnosability of digital circuits. *Proceedings of the 12th International Conference On VLSI Design*. January 10-13, 1999. Goa, India: IEEE. 629-634.

18. Rahagude, N., Chandrasekar, M., and Hsiao, M. S. DFT+ DFD: An Integrated Method for Design for Testability and Diagnosability. *19th IEEE Asian Test Symposium (ATS).* December 1-4, 2010. Shanghai, China: IEEE. 218-223.

19. McDonald, J. Optical Microscopy. In: Ross, R. J. *Microelectronics Failure Analysis Desk Reference, 6th edition*. Materials Park, Ohio, USA: ASM International. 457-476; 2011.

20. Polonsky, S., Jenkins, K. A., Weger, A., and Cho, S. CMOS IC diagnostics using the luminescence of OFF-state leakage currents. *Proceedings of International Test Conference (ITC)*. October 26-28, 2004. Charlotte, North Carolina, USA: IEEE. 134-139.

21. Kash, J. A., and Tsang, J. C. Dynamic internal testing of CMOS circuits using hot luminescence. *Electron Device Letters*, 1997. 18(7): 330-332.

22. Chen, Y.-C. S. and Bockelman, D. Increasing the on-die nodal observability and controllability use of advanced design for debug circuit features. *IEEE International Symposium on VLSI Design, Automation and Test*. April 23-25, 2008. Hsinchu, Taiwan: IEEE. 13-16.

23. Song, P., Stellari, F., Xia, T., and Weger, A. J. A novel scan chain diagnostics technique based on light emission from leakage current. *Proceedings of International Test Conference (ITC)*. October 26-28, 2004. Charlotte, North Carolina, USA: IEEE. 140-147.

24. Bockelman, D., Rahman, A., Wan, I., Chen, S., & Ettinger, S. Multi-Point Probing on 65nm Silicon Technology using Static IREM-based Methodology. *Proceedings of the 31st International Symposium for Testing and Failure Analysis (ISTFA).* November 6-10, 2005, San Jose, California, USA. ASM International. 40-45.

25. Andreev, K., and Racke, H. Balanced graph partitioning. *Theory of Computing Systems*, 2006. 39(6): 929-939.

26. Alpert, C. J. and Kahng, A. B. Recent directions in netlist partitioning: a survey. *Integration, the VLSI journal*, *19*(1), 1-81.

27. Karypis, G., Aggarwal, R., Kumar, V. and Shekhar, S. Multilevel hypergraph partitioning: Application in VLSI domain. *Proceedings of the 34th Design Automation Conference (DAC)*. June 9-13 1997. Anaheim, California, USA: IEEE. 526-529.

28. Chamberlain, B. L. Graph partitioning algorithms for distributing workloads of parallel computations. *University of Washington Technical Report UW-CSE-98-10*, 1998. 3.

29. Pothen, A. Graph partitioning algorithms with applications to scientific computing. *ICASE LaRC Interdisciplinary Series in Science and Engineering*, 1997. 4:323-368.

30. Caldwell, A. E., Kahng, A. B. and Markov, I. L. Improved algorithms for hypergraph bipartitioning. *Proceedings of the 2000 Asia and South Pacific Design Automation Conference*. January 25-28, 2000. Yokohama, Japan: ACM. 661-666.

31. Karypis, G., and Kumar, V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*. 1998. 20(1): 359-392.

32. Karypis, G., and Kumar, V. A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. *University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center, Minneapolis, MN*. 1998.

33. Loewer, J. and Ade, R. tDOM-A fast XML/DOM/XPath package for Tcl written in C. *Proceedings of First European Tcl/Tk User Meeting*. June 15-19, 2000. Hamburg, Germany.

34. Liao, J. Y., Kasapi, S., Cory, B., Marks, H. L., and Ng, Y. S. Scan chain failure analysis using laser voltage imaging. *Microelectronics Reliability*, *50*(9), 1422-1426.