DESIGN FOR PRE-BOND TESTABILITY IN 3D INTEGRATED CIRCUITS

A Dissertation
Presented to
The Academic Faculty

By

Dean L. Lewis

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
in the
School of Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
December 15, 2012

DESIGN FOR PRE-BOND TESTABILITY IN 3D INTEGRATED CIRCUITS

Approved by:

Dr. Hsien-Hsin S. Lee, Advisor
Associate Professor, School of ECE
Georgia Institute of Technology

Dr. Richard Vuduc
Assistant Professor, College of Computing
Georgia Institute of Technology

Dr. Muhannad Bakir
Associate Professor, School of ECE
Georgia Institute of Technology

Dr. Sudhakar Yalamanchili
Professor, School of ECE
Georgia Institute of Technology

Dr. Sung Kyu Lim
Associate Professor, School of ECE
Georgia Institute of Technology

Date Approved: July 27, 2012

To my amazing, wonderful, loving wife Heather.

ACKNOWLEDGMENTS

The completion of a dissertation is a long and arduous process, and I never could have done it alone. Here I would like to thank the many, many people who have helped and supported me along the way.

First, my thanks to my adviser, Professor Hsien-Hsin S. Lee for giving me the opportunity to study under him, for assisting and guiding my research, and for ensuring my success as a scientist, engineer, and academic. I also want to thank Professor Sung Kyu Lim and Dr. Gabriel Loh for their guidance and mentorship in the 3D research group.

My thanks to Professor Muhannad Bakir, Professor Richard Vuduc, and Professor Sudhakar Yalamanchili for lending me their time and insight as members of my dissertation committee. My thanks to Professor David Keezer and Professor Linda Milor as well for serving on my proposal committee.

My thanks to the current and former members of the MARS lab for their years of insight, mentoring, and friendship: Dr. Nak Hee Seong, Dr. Richard Yoo, Manoj Athreya, Ali Benquassmi, Andrei Bersatti, Nishank Chandawala, Eric Fontaine, Jen-Cheng Huang, Ilya Khorosh, Tzu-Wei Lin, Mohammad Hossain, Fayez Mohamood, Lifeng Nai, Ahmad Sharif, Guanhao Shen, Vikas Vasisht, and Sungkap Yeo. A special thanks to Dr. Mrinmoy Ghosh and Dr. Dong Hyuk Woo for their friendship and for leading-by-example in their own Ph.D. studies.

My thanks to the current and former members of the GTCAD lab: Krit Athikulwongse, Rohan Goel, Moongon Jung, Young-Joon Lee, Chang Liu, Pratik Marolia, Shreepad Panth, Mohit Pathak, Hemant Sane, and Taigon Song. A special thanks to Dr. Michael Healy, Dr. Dae Hyun Kim, and Xin Zhao for their friendship and their crucial assistance in the development and publication of my research.

My thanks to the members of the 3D-MAPS design team for the unique opportunity to create a working microprocessor with them. In additional to those mentioned above, my

thanks to Gokul Kumar and Minzhen Ren for this special project.

My thanks to the Georgia Tech community and the wonderful environment they have created there. A special thanks to Dr. Tapobrata Bandyopadhyay, Professor Jeff Davis, Dr. Carl Gray, Dr. Chris Lee, Professor Milos Prvulovic, Dr. Samantika Subramaniam, Pam Halverson, Andrew Kerr, Beverly Scheerer, and Jeff Young. A special thanks to my close friends Dr. Demijan Klinc, Dr. Matthew Lynch and wife Lisa, Dr. Ioannis Doudalis, Professor Guru Venkataramani, and Dr. Kiran Puttaswamy for their companionship, insight, banter, and guidance.

My thanks to the global research community, including conference attendees, reviewers, and funding agencies. A special thanks to Dr. Eric Jan Marinissen for his guidance, feedback, and encouragement.

My thanks to the VCU School of Engineering, the Chesterfield County Math and Science Highschool, and the countless friends and mentors I have had along the way. A special thanks to Lee Adcock, Vivek Agarwal, Jake Bono, Jeremy Davis, Lorie Ros Jacob, Jeff McBride, Jason Naggles, Eric Reisinger, Sarah Rigsbee, Kerry Rose, Brandon Saunders, Julie Wald, and Grant Withers. My life would be unrecognizable without them.

My thanks to my family for their undying support. I am especially thankful for my grandparents, Roy and Marie Lewis and Freeland and Norma Young, four wonderful people who made me who I am, and for my brother Geoff for toughening me up along the way. And I cannot even properly describe my love and my appreciation for my parents, Roy and Jane Lewis. None of this would have been possible without their unwavering love, support, guidance, friendship, and encouragement. From them I learned the value of hard work, commitment, and a kind heart, and to them I owe everything.

Finally, my thanks and my heart to Heather, my wife, for her endless love, support, encouragement, and devotion. Heather, I love you!

TABLE OF CONTENTS

ACKNOWLEDGMENTS iv

LIST OF TABLES viii

LIST OF FIGURES ix

SUMMARY xii

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 ORIGIN AND HISTORY OF THE PROBLEM 3
- 2.1 Design for Test 3
- 2.2 3D Integration 7
- 2.3 3D Testing 9

CHAPTER 3 PRE-BOND TEST ARCHITECTURE AND APPLICATION . . 11
- 3.1 Requirements 11
- 3.2 Hardware 13
- 3.3 Experiments 16
- 3.4 3D-MAPS Test Architecture 19
- 3.5 Summary 32

CHAPTER 4 3D CIRCUIT DESIGN FOR PRE-BOND TEST 34
- 4.1 3D Circuit Design and Test 34
- 4.2 Experiments 40
- 4.3 Summary 45

CHAPTER 5 3D TEST WRAPPERS 46
- 5.1 Problem Definition 47
- 5.2 Wrapper Design Algorithm 50
- 5.3 Experiments 55
- 5.4 Summary 62

CHAPTER 6 SHORTING PROBE 63
- 6.1 3D Via Defects 64
- 6.2 3D Via Probing 65
- 6.3 Experimental Setup 74
- 6.4 Results 76
- 6.5 Physical Considerations 89
- 6.6 Summary 94

CHAPTER 7 RECENT DEVELOPMENTS 95
7.1 IEEE P1838 Standard 95
7.2 Pre-bond Test 96
7.3 Post-bond Test 97
7.4 3D Assembly 99
7.5 3D Via Repair 100

CHAPTER 8 CONCLUSION 102

REFERENCES 104

LIST OF TABLES

Table 1 List of buses in 3D 21264 layout 18

Table 2 Global control signals 27

Table 3 Physical design costs of 3D adder 43

Table 4 Adder power and performance results 43

Table 5 Test vectors required for Kogge-Stone adder designs 44

Table 6 Two-tier circuit benchmarks 54

Table 7 Four-tier circuit benchmarks. 55

Table 8 Stitching wire reuse results 59

Table 9 Shorting probe circuit parameters 75

Table 10 Sensitivity analysis variables 76

Table 11 Monte Carlo statistical results 88

LIST OF FIGURES

Figure 1 Digital logic operational modes 4
Figure 2 3D integrated die stack 7
Figure 3 3D architectural partitioning 12
Figure 4 3D scan-based test architecture 14
Figure 5 3D test-aware clock tree design 15
Figure 6 3D test-aware power rail design 16
Figure 7 Floorplan for 3D 21264 case study 17
Figure 8 The 3D-MAPS chip stack 19
Figure 9 SEM image of the 3D-MAPS chip stack 20
Figure 10 Sector test architecture 21
Figure 11 3D-MAPS single core architecture 24
Figure 12 CTC circuit diagram 25
Figure 13 TCSM state transition diagram 26
Figure 14 3D test path schematic 30
Figure 15 Screen capture of a 3D-MAPS test vector 31
Figure 16 Kogge-Stone adder schematics 35
Figure 17 Register file schematics 37
Figure 18 Flowchart of the 3D register file test algorithm 39
Figure 19 Layouts for a 64-bit Kogge-Stone Adder 40
Figure 20 Layouts for a 1kb register file 41
Figure 21 Motivation for the 3D wrapper design problem 48
Figure 22 Visual description of 3D wrapper design algorithm 50
Figure 23 Psuedo-code description of KL algorithm 52
Figure 24 CTL results for circuit 1 57
Figure 25 CTL results for circuit 2 57

Figure 26 CTL results for circuit 3 58
Figure 27 CTL results for circuit 4 58
Figure 28 Cut results for circuit 1 60
Figure 29 Cut results for circuit 2 60
Figure 30 Cut results for circuit 3 61
Figure 31 Cut results for circuit 4 61
Figure 32 3D via defect scenarios 64
Figure 33 Motivation for shorting probes test methodology 66
Figure 34 MEMS probe tip array 67
Figure 35 Example applications of the shorting probes test methodology 70
Figure 36 Another example application 72
Figure 37 Generalized 3D via assignment plan 73
Figure 38 Shorting probes circuit model 74
Figure 39 Sensitivity results for driver strength 77
Figure 40 Sensitivity results for driver wire length 79
Figure 41 Sensitivity results for receiver strength 80
Figure 42 Sensitivity results for receiver length 81
Figure 43 Sensitivity results for load strength 82
Figure 44 Sensitivity results for the number of loads 83
Figure 45 Knee-point results for sensitivity analyses 84
Figure 46 Sensitivity results for probe tip capacitance 85
Figure 47 Knee and turn-on point results for probe capacitance 86
Figure 48 Monte Carlo distribution of propagation delays 87
Figure 49 3D via variation sources 90
Figure 50 Sensitivity results for driver contact resistance 91
Figure 51 Sensitivity results for receiver contact resistance 92
Figure 52 Knee and turn-on results for contact resistance with driver 93

Figure 53 Knee and turn-on results for contact resistance with receiver 93

SUMMARY

In this dissertation we describe several DFT techniques specific to 3D stacked IC systems. The goal has explicitly been to create techniques that integrate easily with existing IC test systems. Specifically, this means utilizing scan- and wrapper-based techniques, two foundations of the digital IC test industry.

First, we describe a general test architecture for 3D ICs. In this architecture, each tier of a 3D design is wrapped in test control logic that both manages tier test pre-bond and integrates the tier into the large test architecture post-bond. We describe a new kind of boundary scan to provide the necessary test control and observation of the partial circuits, and we propose a new design methodology for test hardcore that ensures both pre-bond functionality and post-bond optimality. We present the application of these techniques to the 3D-MAPS test vehicle, which has proven their effectiveness.

Second, we extend these DFT techniques to circuit-partitioned designs. We find that boundary scan design is generally sufficient, but that some 3D designs require special DFT treatment. Most importantly, we demonstrate that the functional partitioning inherent in 3D design can potentially decrease the total test cost of verifying a circuit.

Third, we present a new CAD algorithm for designing 3D test wrappers. This algorithm co-designs the pre-bond and post-bond wrappers to simultaneously minimize test time and routing cost. On average, our algorithm utilizes over 90% of the wires in both the pre-bond and post-bond wrappers.

Finally, we look at the 3D vias themselves to develop a low-cost, high-volume pre-bond test methodology appropriate for production-level test. We describe the shorting probes methodology, wherein large test probes are used to contact multiple small 3D vias. This technique is an all-digital test method that integrates seamlessly into existing test flows. Our experimental results demonstrate two key facts: neither the large capacitance of the probe tips nor the process variation in the 3D vias and the probe tips significantly hinders

the testability of the circuits.

Taken together, this body of work defines a complete test methodology for testing 3D ICs pre-bond, eliminating one of the key hurdles to the commercialization of 3D technology.

CHAPTER 1

INTRODUCTION

Test is a constant challenge in the integrated circuit (IC) industry. Manufacturing processes are imperfect, yet customers expect working products, so IC manufacturers must, to the best of their ability, ensure that each part is correct before shipping it. The most prevalent modern test solution in digital systems is *scan* and its derivative technologies, which has been used with great success over the past couple decades to ensure final product quality.

Scan-based IC test is a simple idea: stitch all the internal flip-flops into a scan chain, then use this chain to insert test vectors and recover test responses. This provides direct access to the internal logic, greatly simplifying and expediting the testing process. From this basic idea, an entire field of research and development has arisen and lead to key innovations such as built-in self-test, memory self-test, test-time optimization algorithms, black-box-IP self-test, and analog and mixed-signal test. All these are built upon the foundation of scan test.

Underpinning the effectiveness of scan testing is a set of basic IC features, elements of digital IC designs that are critical to execution of a scan test. Some of these features include

- Connected and operational signal nets (i.e., each net has at least one driver and one receiver)
- Connected and operational master signals such as clock and reset
- Connected and operational power and ground rails
- Large off-chip bonding pads for test access

Unfortunately, when we consider the application of scan test to 3D integrated IC chip stacks, we find that many of these basic features are missing within the unbonded dies. All 3D signal nets will necessarily be missing either the driver or the receiver pre-bond,

breaking the test paths; with highly-optimized 3D designs, master signals are fragmented and useless pre-bond; and the large off-chip bond pads exist only on the top tier and so are unavailable to all other tiers pre-bond. In fact, the only feature listed above that can be counted upon is the power and ground rails, which are so ubiquitous in every IC that they remain fully connected even in partitioned 3D designs.

For the rest, new *design-for-testability* (DFT) structures are required to either restore or replace these missing features. DFT is a general design philosophy wherein the ease and effectiveness of product test is considered as a primary requirement throughout the design process. In the case of 3D ICs, the requirements of pre-bond test must be considered from the outset. An unbonded 3D tier is a completely unique target device, unlike any before it. This is because, at the most fundamental level, an unbonded tier is a broken device; part of the basic circuit functionality is located on the neighboring tiers, not on the tier-under-test. This necessitates new testable designs that are specific to 3D IC stacks, and this is the challenge we take up in this book.

The remainder of this book is organized as follows. Chapter 2 presents the details of the 3D test problem and the prior art that forms the foundation of the DFT solutions presented later. Chapter 3 describes a new 3D-aware test architecture and demonstrates its application to a real 3D IC design. Chapter 4 describes extensions to this test architecture for circuit-partitioned 3D designs. Chapter 5 describes a new tool that extends test wrappers, a very successful DFT technique used in planar SOCs, into the third dimension. Chapter 6 describes a brand new technique for testing the 3D structures themselves pre-bond. Chapter 7 summarizes recent developments from other research groups in the field of 3D test. Chapter 8 concludes.

CHAPTER 2

ORIGIN AND HISTORY OF THE PROBLEM

The problem we will study in this book is a product of the collision of two fields: 3D integration and testable circuit design. 3D integration is an exciting new manufacturing technology in which multiple silicon chips are stacked vertically to decrease communication distance while increasing total silicon area. However, it creates significant challenges for test, especially in the unbonded tiers. We will examine both fields in turn.

2.1 Design for Test

Manufacturing is an errant activity, no matter the industry, and it generally makes good economic sense to test products to ensure final quality (that is, the percentage of working parts out of all product shipped). Due to the incredible complexity of modern ICs—just a single stage in a current generation processor might have 2^{128} possible states—designing chips for testability is a basic necessity. The field of *design for test* (DFT) got its start in the 1970s as IC complexity pushed into large and then very-large scale integration. We will examine the key milestones in the development of DFT here.

2.1.1 Scan Test

The most fundamental concept in DFT is scan. The idea is to give an IC two operational modes, *functional* and *scan,* as shown in Figure 1. Functional mode is the normal operative mode of the chip, where it performs the task for which it is designed. Scan mode is the test mode of the chip, where all components are, ideally, reduced to two sets: combinational logic and scan registers. By scanning data into all the registers, the tester gains complete and immediate control of the entire system state, significantly reducing the complexity of test.

Of course, scan is not a complete answer to the IC test problem, for two primary reasons. First, the bandwidth of a scan chain is very limited. Second, only a subset of circuit

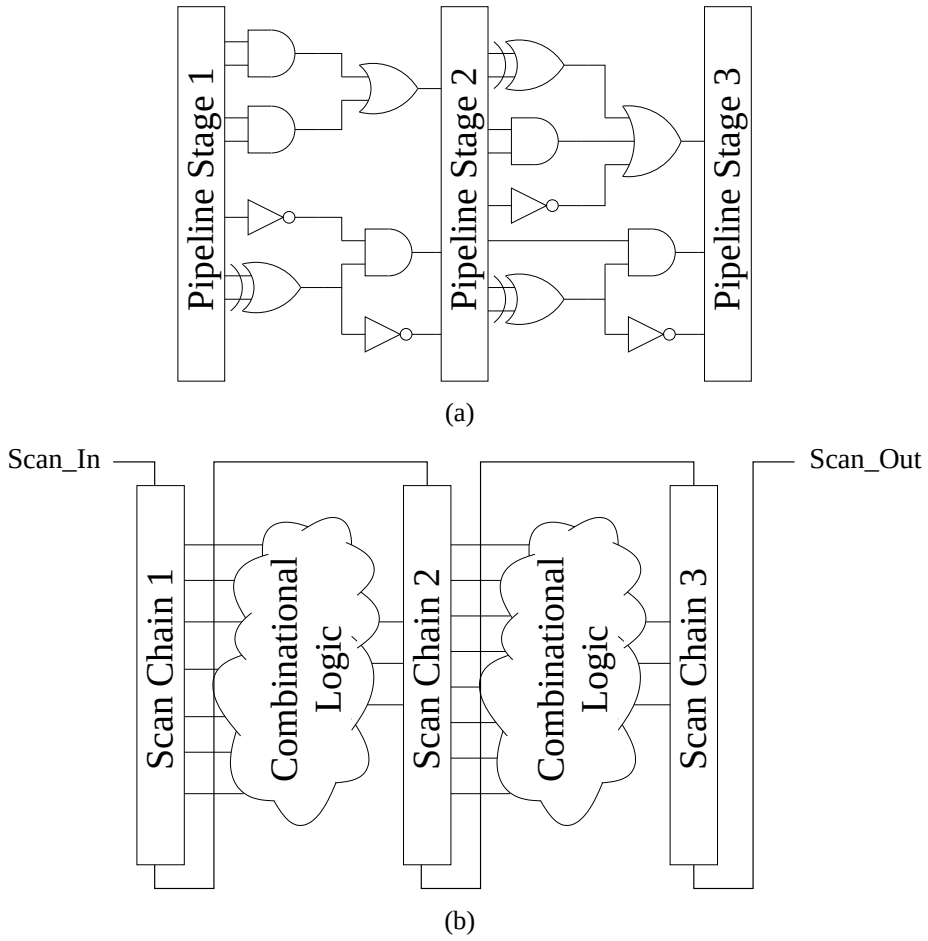


Figure 1. The two operational modes in a simple IC with DFT:(a) functional mode and (b) test mode.

design techniques and technologies can be fit into the ”combinational logic or scan register” abstraction. To address these short-comings, *built-in self test* (BIST) techniques have been developed.

Limited scan bandwidth is generally addressed with a combination of parallelization and test data compression. Unfortunately, the former is limited by the number of package pins available, so even modern ICs have only a few dozen parallel scan paths [51, 70, 77]. More significant test time reductions come from test compression. A BIST implementation of test compression most often relies on linear feedback shift registers to create *pseudo-random pattern generators* (PRPGs) and *multiple input signature registers* (MISRs). PRPGs generate random test data to drive the *circuit under test* (CUT), and

MISRs compress the CUT response into a signature. With these components, the tester needs only to scan in the initial PRPG state and scan out the final signature for comparison, reducing test data load many-fold. The *built-in logic block observer* [39] (BILBO) is perhaps the most complex implementation. It combines the functions of a register stage, scan chain, PRPG, and MISR into a single bank of flip-flops.

For circuits that do not fit the scan abstraction, more application-specific test designs are necessary. The most important, and so most studied, class of these circuits is memory, leading to a entire subfield of BIST research called *memory BIST* (MBIST). Memory represents unscannable IC state, so MBIST techniques must work with the addressing features of the memory system to successfully execute memory test. This generally consists of a carefully-designed pattern of reads and writes to activate various possible faults. Two example sequences are the *Algorithmic Test Sequence* [38] (ATS) and *Galloping 1's and 0's* [13] (GALPAT) though there are certainly many more [6]. ATS detects all stuck-at faults in a memory, while GALPAT extends this fault coverage to include all coupling faults between memory cells as well. Most MBIST algorithms range between $O(n)$ and $O(n^2)$ complexity, where n is the number of memory cells; applying that many patterns one at a time through scan is simply economically impossible for any reasonably-sized memory but is very feasible with BIST.

2.1.2 Modular Test

Of course, verifying the operation of the component ICs is not sufficient to guarantee a working computer system. The motherboard and other PCBs are also critical. Originally, PCBs were tested with probes. The tester would touch each end of a PCB wire with probes to verify it was manufactured properly. But this is not cost-effective in modern PCBs which can have many thousands of wires. To address this problem, the *boundary scan register* (BSR) was developed in the 1980s and formalized in 1990 as the IEEE 1149.1 standard [3]. The BSR is just a scan chain which contains a scan cell for every signal pin in or out of an IC. To test a PCB bus between two ICs, the manufacturer needs only to scan

the test data into the BSR of one IC and then read that same data out of the BSR of the other IC. Most importantly, the IC vendors do not need to surrender any of their IP to the PCB manufacturer to enable this test (other than a description of the BSR).

The 1149.1 standard also describes a *test access port* (TAP)—the TAP chiefly contains a state machine, a command register, and multiplexers—that must be used to interface the BSR to the PCB's test architecture. With this TAP in place, IC vendors realized they could also use it to access internal IC test features after system integration. This enabled the vendors to ship test bit streams with their products. The PCB manufacturer could then apply these bit streams to the TAP and verify the correctness of the IC, all without knowing the actual details of the IC. This created a robust system of modular testing (or *black box testing*) of components to verify the final product.

A collection of test resources such as those defined in 1149.1 is known as a *test wrapper*. The chief function of a test wrapper is to create boundaries within the test architecture for isolating different modules from one another, allowing them to be tested independently. 1149.1 test wrappers, for example, allow ICs and the buses that interconnect them to all be tested independently.

With the advent of *systems-on-chip* (SOCs) and other products of similar complexity, the concept of the test wrapper was adapted to in-chip test as well. Now instead of isolating ICs and PCB buses from one another, the goal is to partition the chip itself into several modules that can then be tested individually. In a true SOC, the IP blocks define a natural partitioning scheme; in monolithic ICs, chip functionality can define the scheme (for example, isolating the processing core from the various units of the memory hierarchy). Test wrappers for SOCs were formalized in 2005 in the IEEE 1500 standard [4].

Adapting test wrappers to SOCs was not straightforward. Because of the limited amount of data required to test buses between ICs, the 1149.1 standard calls for a single one-bit test data bus. The 1500 standard however was designed from the start for both testing the buses between IP blocks and for testing circuits internal to the IP. This requires a much greater

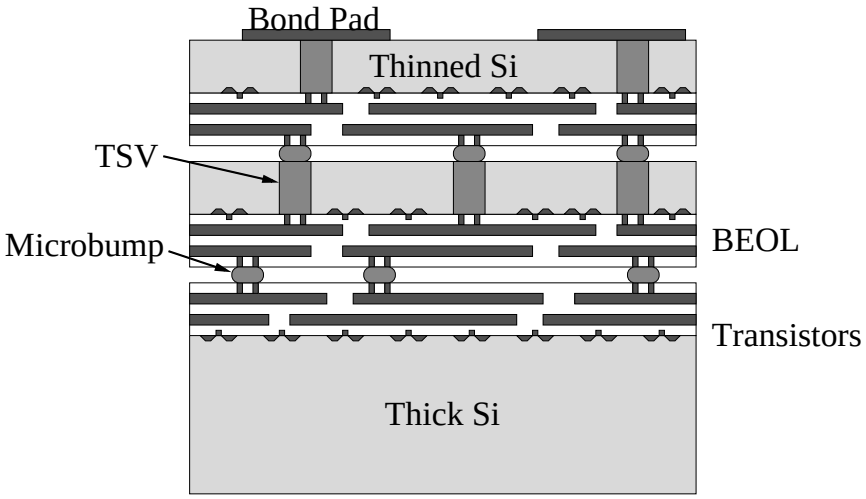


Figure 2. A 3D integrated die stack.

test data volume. Therefore, the test architecture problem expanded to include the design of the *test access mechanism* (TAM), the multi-bit test bus used to interconnect the test data ports on each test wrapper. An SOC test architecture may include just one wide TAM or many skinny TAMs, whichever minimizes the test cost. This makes the effort of designing a test architecture for 1500 test wrappers much more difficult than for 1149.1 test wrappers.

Now this cycle of modular test design must begin again, this time for 3D ICs.

2.2 3D Integration

3D integration (shown in Figure 2) is an emerging technology that allows semiconductor die to be bound together to form a tightly integrated stack. Opening design to the third dimension provides several advantages. First, it enables the integration of heterogeneous components such as logic and DRAM memory [11, 71] or analog and digital circuits [9]. Second, it increases routability [65]. Finally, it can substantially reduce wire length, which contributes to both long communication latency and to high power consumption. Recent work in this field has already demonstrated significant improvements in both performance and power consumption [69, 83] and lead to other interesting applications, such as on-line profiling [54] and network-in-memory [43], and even greater returns are expected as

researchers further explore the opportunities afforded.

2.2.1 Die Stacking

3D stacking replaces the long, heavily-loaded interconnects of present day integration schemes—for example multi-chip modules or package-on-package stacks—with short, fast 3D vias (which may be backside TSVs, faceside microbumps, or a combination of both). 3D via manufacturing lines already exist which can produce vias approximately one micron in diameter, and companies are pushing into the submicron domain, testing 0.4μm 3D vias [64].

2.2.2 3D Partitioning Granularity

Die stack technology may be used to partition a design at three general levels of granularity. The coarsest level is the technology level. Disparate technologies like high-speed CMOS and high-density DRAM both have their own dedicated and highly-optimized manufacturing processes. Many problems arise when attempting to integrate such technologies onto a single die, requiring sophisticated manufacturing tricks to achieve economically viable integration quality [60]. Die stacking allows each technology to be manufactured on its own tier in its own process. After each tier is manufactured, a separate integration process bonds these tiers together. The result is the best of both worlds: each tier is manufactured at the highest possible quality level and, simultaneously, the two technologies are tightly integrated. This improves both the performance of the system and the form factor.

The next finer level of partitioning is the architectural level. Unlike technology partitioning, both tiers are manufactured using the same process. The goal of architectural partitioning is to spread the functional blocks of a design across the available tiers in such a way as to minimize the length of the interconnect buses. By reducing bus length, the resistance and capacitance seen on these buses is reduced, consequently reducing power consumption and improving performance. Architectural partitioning makes much better use of the large number of 3D vias available than technology partitioning.

The finest partitioning granularity is the circuit level. Here, the transistors that make up a functional block may exist on different tiers. Circuit partitioning has its own levels of granularity. At one extreme, blocks are simply split along logical boundaries into sub-blocks (e.g. a design could place half the banks of a cache on one tier and the other half on a different tier—so called bank-stacking [43, 66]). At the other extreme, individual circuits are split across the tiers (e.g. in a register file, read and write ports may be spread across different tiers, connected to the actual memory inverter pair through 3D vias; this is known as port splitting [69]). This granularity best utilizes the available 3D vias and thus shows the best power and performance improvements.

2.3 3D Testing

The problem we address then is enabling test in a 3D integrated chip stack. There are three different test situations to consider:

1. *Pre-bond* — a single tier is under test which is not bonded to any other tier
2. *Partial-stack* — some incomplete subset of the chip stack is under test, including the bonds between the tiers in this subset
3. *Post-bond* — also *final stack*, the entire completed chip stack is under test

Post-bond test is the least interesting case. Once the chip is complete, all chip components are existent and functional, so the situation is identical to that of bare-die test in traditional planar manufacturing lines. Pre-bond and partial-stack tests are much more interesting and challenging because some of the chip functionality is necessarily missing. Additionally, the 3D vias represent dangling nets, which are a challenge unique to 3D.

To enable pre-bond test[1] then, we require DFT features both to compensate for missing functionality and for establishing controllability and observability over dangling 3D connections. The work presented in Chapters 3, 4, and 5 addresses these issues for the circuitry

[1]Hereafter we refer only to pre-bond test rather than both pre-bond and partial-stack test because both face the same key challenges and benefit from the same solutions.

internal to each tier. The work presented in Chapter 6 presents a methodology for testing the 3D vias themselves pre-bond.

CHAPTER 3

PRE-BOND TEST ARCHITECTURE AND APPLICATION

The overall DFT plan for a chip is called the *test architecture*. The test architecture is the chip-wide master plan the organizes and manages all the various DFT components within the chip and provides an off-chip interface for test execution. It is through the test architecture that the multitude of scan chains, BIST engines, test wrappers, and other test features are accessed.

Generally, test architectures are designed to rely on the correct operation of as few chip features as possible because if the test architecture fails, the chip is effectively worthless, even as a trouble-shooting tool. These features include such things as a working clock, properly-charged power rails, a set of operational control signals (*reset*, *test_enable*, *clk_ctrl*, etc.), and a minimum number of functional I/O pins (usually just four).

By-and-large, these are fairly simple needs, and of course that is the point of designing the test architecture in such a manner. 3D integration, however, adds a new twist to the story, which we will explore in this chapter.

3.1 Requirements

There are several requirements a pre-bond test architecture must meet in order to successfully enable pre-bond test. We examine these requirements and the challenges each addresses here [40].

3.1.1 Completing the Design

The primary testability challenge posed by 3D integration is that, pre-bond, each tier exists in an incomplete state. For a technology partitioning, there is no problem, as each tier is by definition functionally complete. For an architectural partitioning (for example, the partitioning of a processor core shown in Figure 3) however, there are problems. Traditional test methodologies [10, 51, 63, 70, 77] can depend on full connectivity within the chip,

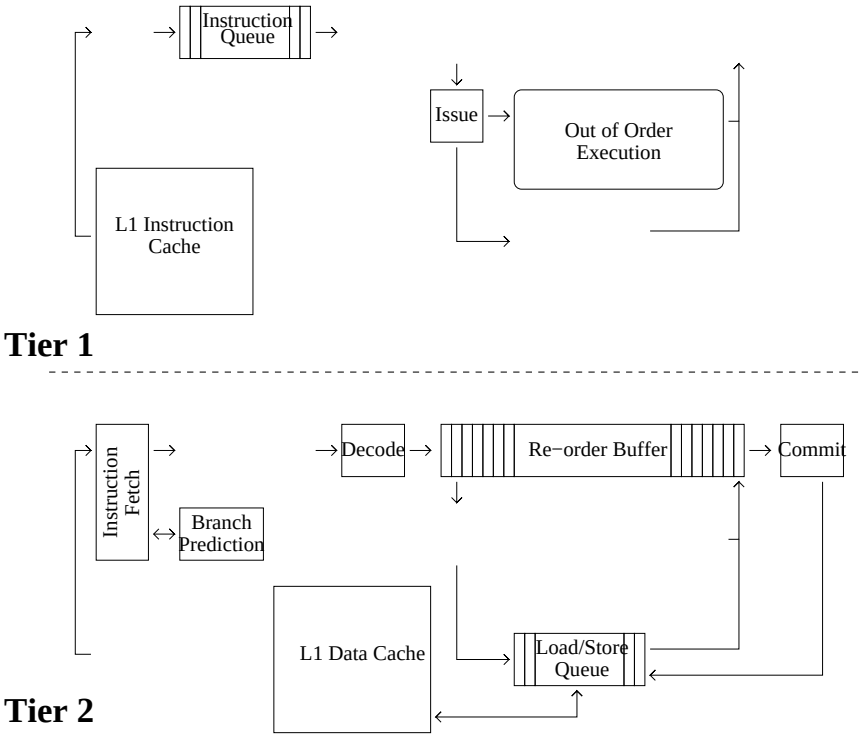


Figure 3. Example partition a generic out-of-order processor across two tiers.

especially in functional or partial-scan test, but this connectivity is not guaranteed in a 3D IC pre-bond. The situation is exacerbated by circuit-level partitionings wherein even the functional blocks are incomplete, and, worse, the circuits themselves may be incomplete and functionally broken. This leads to a paradox of sorts in that we want to test broken circuits to see if they function correctly [41]. Testing circuit-paritioned 3D designs will be discussed in Chapter 4.

The simple brute-force solution would be to probe each 3D via individually, providing or observing test values as necessary. Unfortunately, this will not work for pre-bond test; the number of 3D vias on a given tier can vary from hundreds to hundreds of thousands, and no probe card can provide that many test channels [81]. Therefore, a pre-bond DFT architecture must either replace the missing connections or enable new methodologies for

testing without them.

3.1.2 Test Hierarchy

The pre-bond DFT architecture does not exist in a vacuum. Post-bond test, package test, and so on will follow. Therefore, to keep the cost of test down, the pre-bond test architecture must be designed to integrate with the test architectures for these other methods and provide maximum reuse of test modules.

3.1.3 Hardcore

The *hardcore* of a chip is its infrastructure, nets like `power`, `ground`, `clock`, and `reset` that must be complete and functional for the tier to be able to work by any definition. Any DFT architecture must carefully consider these nets to make sure they are fully connected and operational.

3.1.4 External Access

While the 3D vias cannot be individually probed, some sort of external access via test probes and pads is required to both power the tier hardcore and provide the test access. In all but the top tier of the die stack, these pads must simply be buried post-bond. Thus, the DFT architecture must use this resource very judiciously to control the area cost.

3.2 Hardware

Here we present our 3D DFT architecture and examine how it meets the requirements laid out.

3.2.1 Tiers as Test Modules

As mentioned previously, modular test with test wrappers is a very popular technique. Independent test modules has been successfully applied in many large ICs; an example is the Alpha 21364, which was partitioned with test wrappers into what the Alpha team called *scan islands* [10]. Data would flow freely between islands in functional mode. In test mode however, the test wrappers closed the borders between islands, replacing the

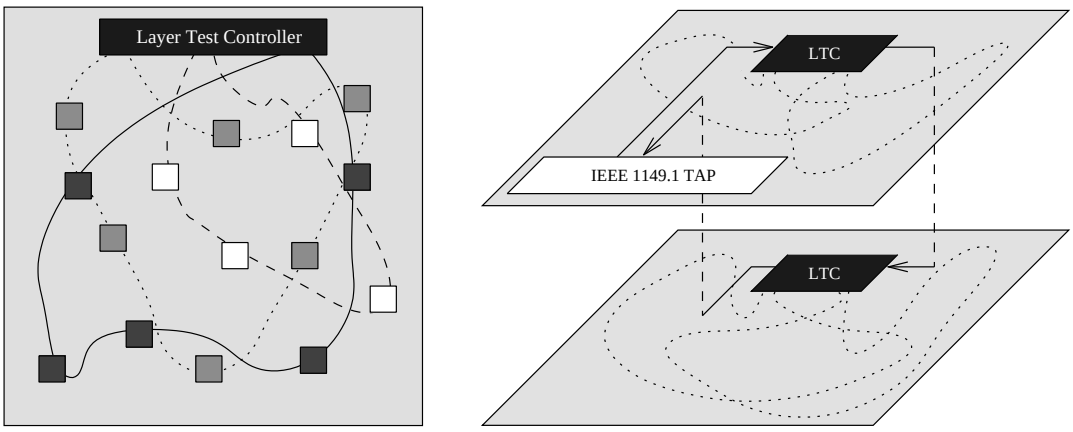


Figure 4. Our 3D test architecture. (a) shows a single tier with connections from the LTC to the various scan chains. (b) shows the LTCs integrated into the chip-level test architecture.

functional signals with test signals from the scan chain. By segmenting the design into several testable modules, such designs significantly reduce the cost and complexity of test.

Comparing this approach to 3D designs, it is clear that each tier, before bonding, exists as a perfectly isolated test module—a condition the Alpha designers were not able to achieve. Thus we adopt this general test strategy to design our pre-bond test architecture, essentially enclosing each tier in its own test wrapper. The central feature of these tier wrappers is the *Layer Test Controller* (LTC), which manages access to the scan chains on the tier (or to lower-level test wrappers if they are in use). Figure 4(a) shows a generic 3D tier with scannable registers hooked up into three scan chains controlled by an LTC. Note that scan cell ordering is well-studied problem [8, 12, 28, 47] and so is not considered here.

Critically, the LTC patches nicely into next higher-level wrapper in the test hierarchy (Figure 4(b)). This satisfies our second requirement and allows for the resources created for pre-bond test to be reused in subsequent test.

To complete our test architecture, the dangling 3D nets must be tied off. As in the prior art, we accomplish in most cases by inserting boundary scan cells as appropriate, satisfying the primary requirement a pre-bond test architecture. These scan cells are necessarily gated so that they do not compete with the 3D-via-connected sources post-bond.

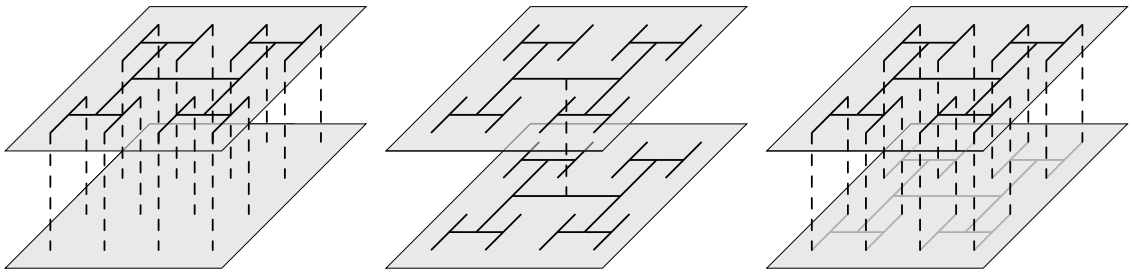


Figure 5. Three 3D clock trees. (a) is optimized for wire length and power consumption while (b) is optimized for pre-bond testability. (c) is the best of both.

3.2.2 Tier Hardcore

None of the features of our test architecture are of any use without the test hardcore, the third pre-bond test requirement. The `power` and `ground` rails are not a concern. These rails are so ubiquitous and so heavily utilized that they will always be fully connected in every tier. This observation is confirmed by the 3D-MAPS test chip [27].

This is not so for other hardcore signals, generally any signal such as `clock` or `reset` which are wire length limited. These nets benefits greatly from 3D design, significantly reducing wire length and power consumption [53]. Figure 5(a) shows an H-tree design for clock distribution in a 3D chip stack. Note that the tree exists almost entirely in the upper tier while 3D vias provide local clock connectivity on the bottom tier. This greatly reduces the cost of the clock, but the many small clock trees on the bottom tier are completely useless for pre-bond test.

An alternative, test-friendly clock tree is shown in Figure 5(b). The clock is fully connected on every tier and so can be used for pre-bond test. However, the cost of the clock is much greater in this design because of the large amount of redundancy in the distribution network.

Our solution is a hybrid design as shown in Figure 5(c). This design is comprised of a 3D-optimized main clock tree (in black) and a pre-bond test tree (in gray). Not shown are tri-state buffers which must be located at each leaf of the pre-bond tree to disconnect it

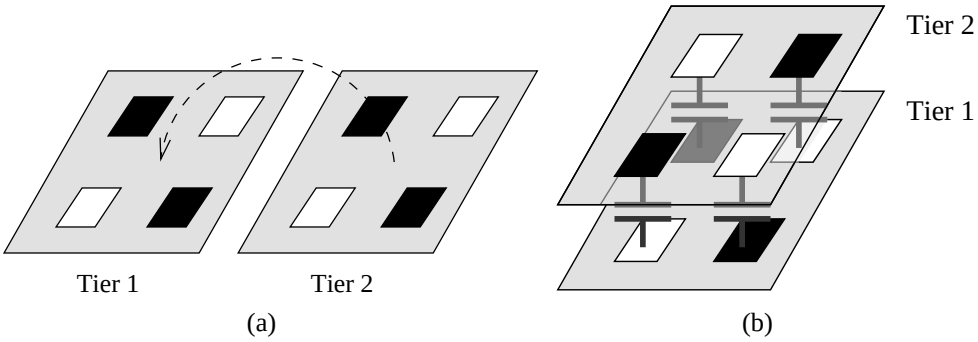


Figure 6. A power rail test pad arrangement that is post-bond reuse aware.

post-bond. Such a design both enables pre-bond test but also saves clock power post-bond, at the cost of bottom-tier routing resources. For stacks greater than two, a pre-bond tree is necessary for each tier. Hybrid 3D clock trees were fully evaluated by Xin et al. in [85, 86]. They created a CAD tool to design these trees and reported power savings around 20%.

3.2.3 External Access

Probe pads, as stated, are unavoidable. The use of test wrappers significantly reduces the number of pads required (the LTC requires a similar test access width as 1149.1 and 1500, four signals minimum). But to simply bury these pads post-bond is wasteful. We propose reusing them as *decoupling capacitors* (decap) as shown in Figure 6. If the pads are already tied to power and ground rails, nothing more is required than to line them up (the pad pattern shown is recommended since the same probe card can be used for each tier). If the pads are tied to other signals, a simple fuse or similar circuit element can tie them to one of the rails post-bond.

3.3 Experiments

3.3.1 Architectural Partitioning

Our experiments are based on the architecture and technology of the Alpha 21264. In order to evaluate the cost of implementing our pre-bond test strategy, we need to know the area consumed by a scan cell and the number of scan cells required in a 3D-integrated design.

To determine a realistic size for the scan cell, the scan cell was laid out using 0.25μm

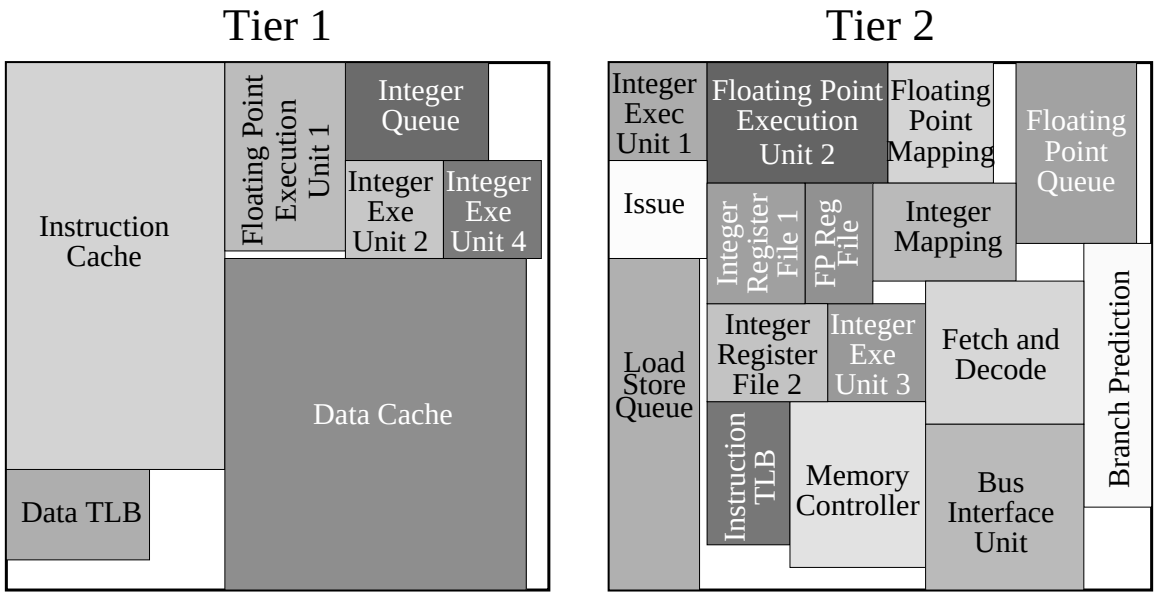


Figure 7. A floorplan for a two-layer die stack split by architectural block. The gray areas between and around blocks represents whitespace within the floorplan.

TSMC design rules. This technology generation was selected to match, as closely as possible, that used to manufacture the 21264A. The actual design of the scan cells is based on the 8T latch. Each cell requires 75.8μm^2 of silicon.

To determine the number of cells required by our technique, a sample 3D floorplan (Figure 7) for a 21264 was designed by a published 3D floorplanner [82]. From this floorplan we extracted the number of signals crossing between the die. Table 1 lists all of the inter-die buses, the number of signals comprising that bus, and the cost of adding the necessary scan cells. Note that each signal requires two scan cells: one on the source side to observe the test output and another on the sink side to provide a test input.

The bottom row in Table 1 gives the final area cost of injecting and observing test values on 3D signals. This cost is 0.165% of the area of the sample floorplan in Figure 7. However, the floorplan contains 8.56% whitespace, so the scan flops do not require an expansion of the chip footprint. Additionally, the area consumed by the scan flops is only 0.173% of the die size of the original Alpha 21264A, which results in a negligible expansion of the die footprint.

Table 1. This list consists of the buses that cross from one tier to another. Listed are the source block and tier, the sink block and tier, the number of signals, and the area penalty paid to include scan flops.

SOURCE	Tier	**SINK**	Tier	**BITS**	**AREA** (μm^2)
Instruction Cache	1	Instruction TLB	2	40	6065
Instruction TLB	2	Instruction Cache	1	174	26384
Instruction Cache	1	Fetch and Decode	2	128	19409
Fetch and Decode	2	Instruction Cache	1	42	6369
INT Mapping	2	INT Queue	1	200	30326
INT Queue	1	Issue	2	196	29720
INT Register File 1	2	INT Execution Unit 2	1	150	22745
INT Execution Unit 2	1	INT Register File 1	2	71	10766
INT Execution Unit 2	1	INT Mapping	2	14	2123
INT Execution Unit 2	1	Branch Predictor	2	93	14102
INT Register File 2	2	INT Execution Unit 4	1	150	22745
INT Execution Unit 4	1	INT Register File 2	2	71	10766
INT Execution Unit 4	1	INT Mapping	2	14	2123
INT Execution Unit 4	1	Branch Predictor	2	93	14102
FP Register File	2	FP Execution Unit 1	1	154	23351
FP Execution Unit 1	1	FP Register File	2	71	10766
FP Execution Unit 1	1	FP Mapping	2	14	2123
Load/Store Queue	2	Data TLB	1	66	10008
Load/Store Queue	2	Data Cache	1	180	27294
Data Cache	1	Load/Store Queue	2	144	21835
Data Cache	1	Memory Controller	2	166	25171
Memory Controller	2	Data Cache	1	166	25171
TOTAL				2397	363,461

Our experiments assume a simple LTC design. The LTC provides parallel access to sixteen scan chains per layer. Additionally, the LTC contains sixteen one-bit bypass registers. Finally, sixteen multiplexers and demultiplexers are included to allow selection between the scan chains and the bypass registers. Together, this allows for sixteen scan chains per layer—thirty-two chains in the chip—which is comparable to modern designs [70]. This design requires thirty three test pads per layer: $S_i[15,0]$, $S_o[15,0]$, and a select signal. The area cost of such an LTC is insignificant compared to the cost of the injection and observation scan cells.

This area cost represents the worst-case cost we should expect for implementing this test technique for two reasons. First, academic layouts produced under publicly available

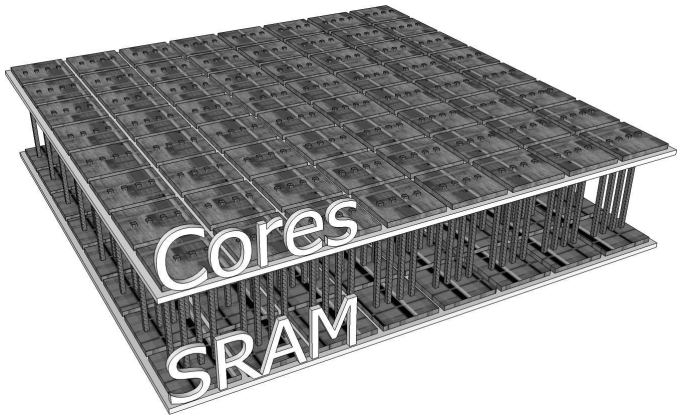


Figure 8. A schematic of 3D MAPS chip stack, showing the sixty-four cores, sixty-four SRAM tiles, and 3D connections.

DRC rules are much larger than functionally-equivalent industrial designs produced under highly-optimized and proprietary DRC rules. Second, we assume a worst-case scan cell scenario in which *every* 3D via requires the addition of two scan cells that serve no purpose beyond pre-bond test value injection or observation. In a real design, many of these cells could be unnecessary—if the 3D via directly sources and/or sinks a scannable flip-flop—or could be reused as part of the post-bond test strategy. For these reasons, we expect an actual application of our technique in an industrial design to cost even less area than the results reported here.

3.4 3D-MAPS Test Architecture

The 3D Massively Parallel Processor with Stacked Memory (3D-MAPS) chip is a test vehicle for evaluating the benefits of 3D fabrication. The design goal was to produce a processor that could consume as much 3D bandwidth as possible and demonstrate the performance improvements expected of applications running on such a system.

The test architecture in the 3D-MAPS chip is based on design-for-pre-bond-test principles that have presented in this chapter, so here we present the details of 3D-MAPS as a case study in pre-bond-testable design.

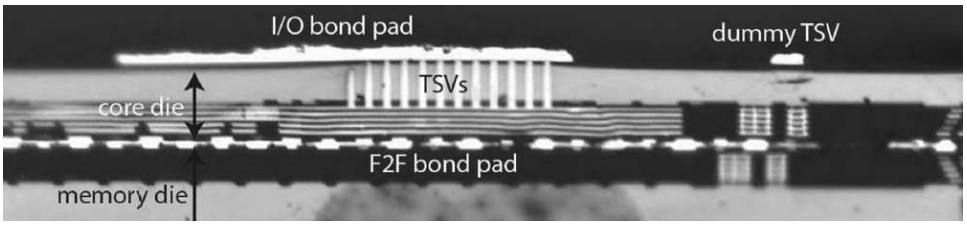


Figure 9. An annotated SEM image of the 3D-MAPS chip showing the key 3D components: backside I/O pads, TSVs, thinned top tier, and the microbump face-to-face bond.

3.4.1 3D Processor Design

3.4.1.1 Chip Stack

The basic architecture of 3D-MAPS is shown in Figure 8 while an image of the actual 3D-MAPS chip stack is shown in Figure 9. The stack consists of two tiers (5mm on a side for 25mm^2 of silicon per tier or 50mm^2 total) bonded face-to-face with microbumps (3.4μmsize, 5μmpitch). Global Foundries [25] fabricated the front-end-of-line (130nm bulk-Si), TSVs (via-first process; 1.2μmsize, 2.5μmpitch), and back-end-of-line (six metals). The thick (765μm) wafers were shipped to Tezzaron Semiconductor for finishing, including bonding (thermo-compression), thinning (12μmtotal, composed of 6μmbulk and 6μmBEOL), I/O pad deposition, and dicing. The I/O pads are placed on the backside of the thinned die (235 I/Os; 14 carry signals, the remainder are power and ground). 204 TSVs are used per I/O cell to handle off-chip current loads.

3.4.1.2 Architecture

3D-MAPS is composed of sixty-four processors and sixty-four SRAM data memories (one private memory per processor). A 116b 3D bus connects each processor to its memory. Each core is a five-stage, in-order, two-wide VLIW machine. The two-instruction format was chosen to maximize utilization of the 3D bus; each core can execute a memory instruction every cycle, for a total 3D bandwidth of 71GBps at 277MHz operating frequency.

Within each core is a 1.5kB instruction memory (192 bundles) and a 1kb register file. Each memory tile is composed of four 1kB memory banks. That is 4kB of data memory per core and 256kB total in 3D-MAPS.

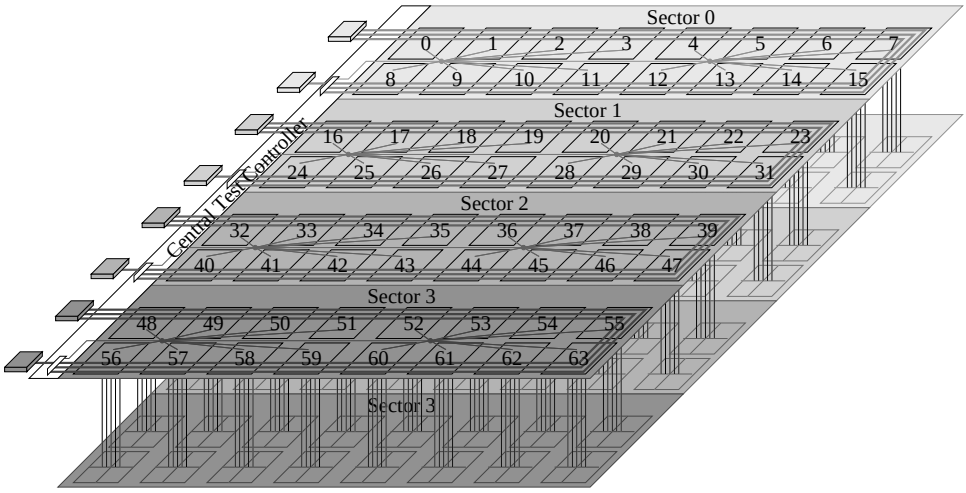


Figure 10. General test sector architecture. Shown are the sixty-four cores divided into four sectors, the twelves scan chains (three per sector), and the 3D interface between the tiers.

For communication, 3D-MAPS has an eight-by-eight mesh network. Each processor can pass data to its four neighbors. This mesh provides 8.9GBps bisection bandwidth. There is no communication between memory tiles; coherency must be maintained by the programmer. A barrier instruction is provided for synchronizing the cores.

3.4.1.3 Off-chip Interface

The functional off-chip interface is limited to three bits, which are physically multiplexed onto the test pins. These three are *done*, *barrier_req*, and *barrier_ack*. *done* signals the end of computation, and *barrier_req* signals that all cores have reached the barrier. Both of these signals are produced by AND trees that reduce the individual *done* and *barrier_req* signals of the sixty-four cores to a single output. *barrier_ack* is a control signal which provides breakpoint-like functionality and discussed further in Section 3.4.2.3.

3.4.2 Sector Test Architecture

The test architecture design process had two goals: graceful degradation and easy experimentation. Graceful degradation is the ability of the design to isolate faulty, failing portions of the chip from good, functional portions of the chip. Graceful degradation is particularly

important to this design because 3D integration is a largely untested manufacturing process, and we need to be able to make measurements with the chip even in the presence of many faults. Easy experimentation is the ability to control and observe the workings of the chip on deep, simple level.

To achieve these goals, we choose a sector-based full-scan test architecture, as shown in Figure 10.

3.4.2.1 Graceful Degradation

A *sector* is a set of sixteen cores which are designed to test and operate independently of all other cores. Each sector is independent from the core level all the way up to the off-chip interface. This provides coarse-grained graceful degradation because a fault within a sector disables only that sector, not the entire chip.

There are a few key aspects to isolating a sector. First, each sector can disable the on-chip mesh network at the boundary of the sector. When the boundary is closed, the sector receives all zeros on that link, rather than faulty communications. This behavior matches the boundary behavior of the full, eight-by-eight mess.

Second, each sector has independent AND-reduce trees for the *done* and *barrier_ack* signals (Figure 10 shows one AND tree in the middle of each sector). In the final stage of reduction, each sector's signal is masked by a sector disable bit. This prevents a faulty sector from interfering with these reductions.

Third, each sector has an independent set of scan chains, as represented by the three thick wires in each sector in Figure 10. No sectors share any part of their scan chains, so a fault in a single scan chain disables only the sector in which that scan chain is found.

Finally, each sector has an independent pair of I/O test pads (shown on the left side of Figure 10. The scan chains for each sector are tied to that sector's I/Os so that even at the off-chip interface, the sectors are independent. Therefore if one of the pads is faulty, only the associated sector is lost; the others can still be subjected to experimentation.

As shown in Figure 10, 3D-MAPS is composed of four sectors. Four was chosen due

to area and pin-count constraints; fewer sectors would have provided too little graceful degradation, and more would have required too much area to implement.

The only hardware shared between the sectors in the hardcore and the test control. The hardcore consists of the power and ground rails, the clock tree, and the reset signal. The test control is composed of the test control state machine (TCSM) and the various enable signals it produces; test control is discussed in detail in Section 3.4.2.3. Isolating this hardware between sectors would have incurred much too high an area and design complexity cost to implement effectively. It is important to note that that communication between the sectors and this shared hardware is one-way;[1] a fault within a sector cannot propagate up through the shared hardware to fail the chip. A fault in the shared hardware itself could fail the chip, but the area of this hardware is quite small and so is an acceptably small failure risk.

3.4.2.2 Easy Experimentation

The other primary goal of the DFT design was ease of experimentation. We need to easily get deep into the chip and observe the various pathways. Most important is the 3D interconnect between the tiers, though general access to all paths is preferred. This is most simply achieved with a full-scan test solution. This provides simple, direct access to all parts of the chip and has greatly eased experimentation. Additionally, we implement some programming chains to control the length of the data-carrying chain.

Figure 11 shows a simple schematic of the single-core architecture. The large circles on the buses into and out of the data memory indicate 3D connections. In particular, this schematic highlights the functioning of the three scan chains. First and most important is the *General Scan Chain* (GSC). This chain snakes through each and every flip-flop in the processor core. This chain contains 772 flip-flops per core (12,352 in a sector). This is the chain that is used to load test vectors and return test responses.

To manage the length of the GSC, we implemented two control chains, the *Pipeline*

[1]The exception is the power and ground rails. A short anywhere in these networks will fail the chip.

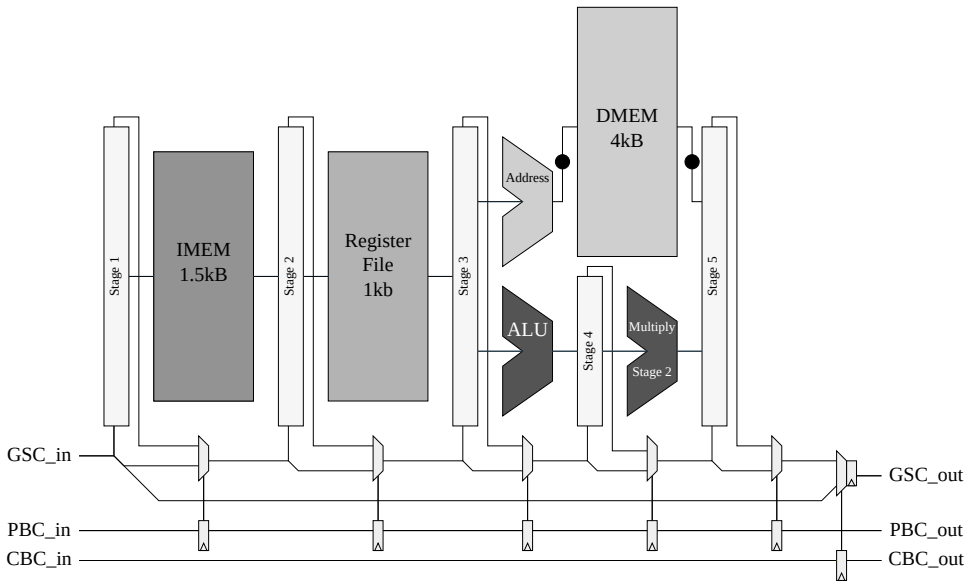


Figure 11. A schematic of the single-core architecture, highlighting the function of the three scan chains.

Bypass Chain (PBC) and the *Core Bypass Chain* (CBC). The PBC is used to exclude individual GSC segments within a core. It is composed of nine bits per core, 144 per sector (four are not shown in Figure 11; they correspond to buffers needed to communicate with a core's four neighbors). The CBC is used similarly, but it bypasses an entire core's GSC segments; the CBC contains 16 bits per sector. Note that the GSC has one unbypassable flip-flop on its output. Its purpose is to prevent timing violations; without it, multiple cores could be bypassed and the GSC could run for millimeters without encountering a flip-flop, which would fail the set-up time requirement.

3.4.2.3 Central Test Controller

The *Central Test Controller* (CTC) is shown in Figure 12. This unit controls all operation (both functional and test) of 3D-MAPS. Because this test chip lacks traditional off-chip memory interfaces, the CTC serves as the only connection between the processor and the outside world. Modeled after the IEEE 1149.1 test access port, the CTC contains some components that are specific to each sector (and so independent from one another as require for graceful degradation) and some shared components.

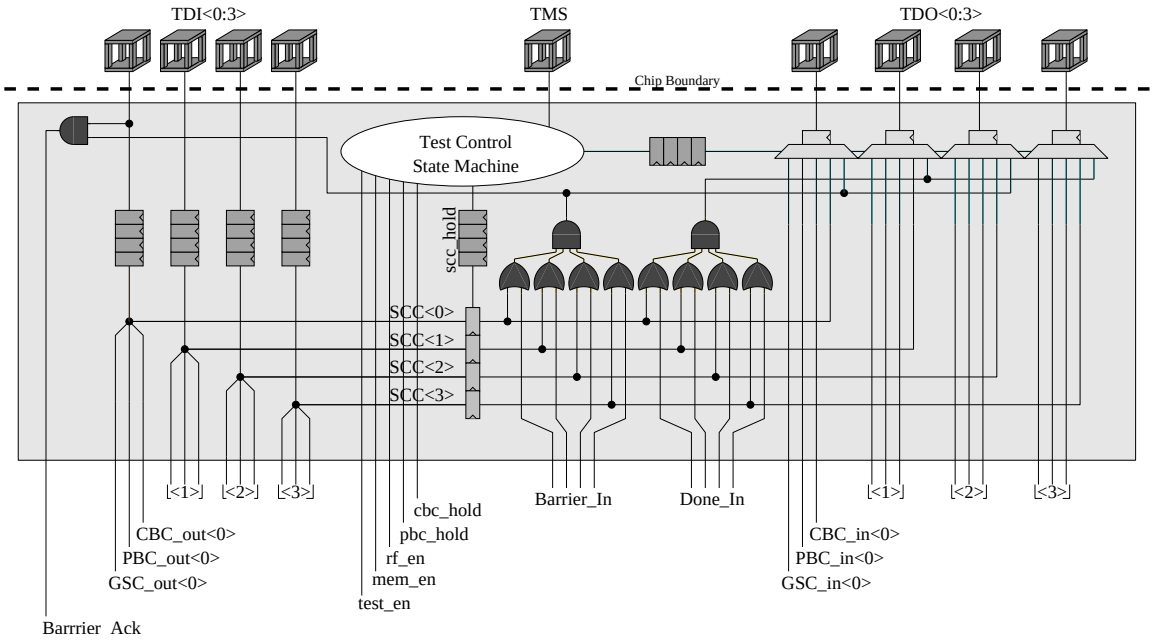


Figure 12. A circuit diagram of entire CTC, including per-sector components.

Sector Hardware Each sector, as was also shown in Figure 10, is given a pair of I/Os. These pins are the *Test Data In* (TDI) and *Test Data OUT* (TDO) pins respectively, and they function to insert data into the processor and capture data produced by the processor. As shown at the top of Figure 12, the TDI and TDO signals (and all other off-chip signals) must traverse the redundant TSV arrays to access the I/O pads on the backside of the thinned tier. Internal to the CTC, the TDI signals are delayed by four cycles; this synchronizes the arrival of the scan chain signals at the first processor core with the arrival of the global control signals produced by the TCSM, which require four cycles to broadcast. The TDO signals have an attached flip-flop as well; this final flip-flop serves to maximize the timing margin available for the output signal to traverse the package and PCB.

Also internal to the CTC is a fourth scan chain for each sector, the one-bit-long *Sector Control Chains* (SCC). The SCC is the bit that actually disables a bad sector, both closing the sector boundaries and masking its *done* and *barrier* signals. Because it is so short, the SCC also serves as a quick way to test the functionality of the CTC itself.

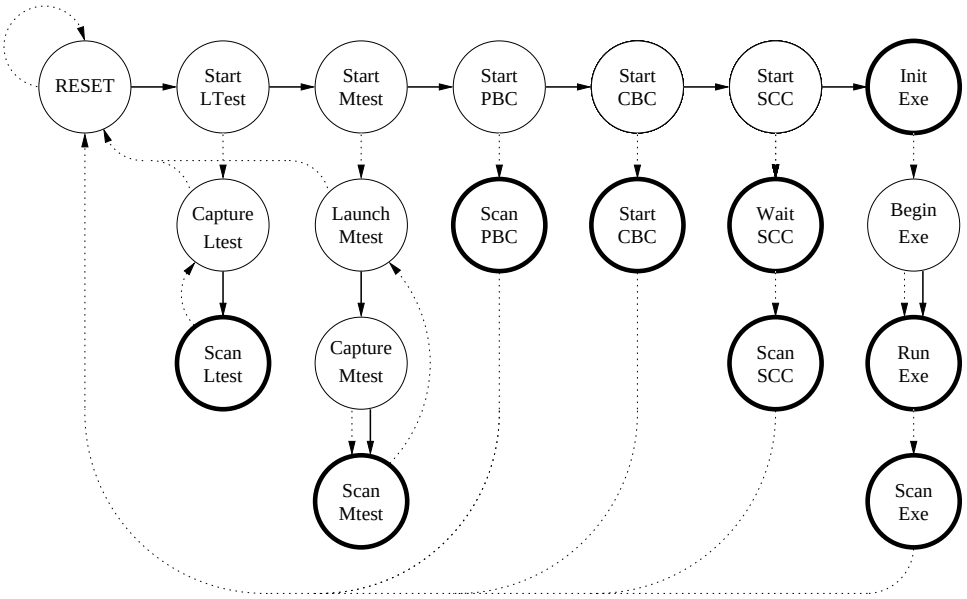


Figure 13. State diagram for the TCSM. Dashed arrows represent TMS='0' transitions, solid arrows TMS='1' transitions. Bolded states do not change state on TMS='1'. Note that holding TMS='0' will always return the machine to the reset state within three clock cycles. For clarity, output signals are not shown; see Table 2.

Test Control State Machine The shared components consist of the TCSM and the *barrier* and *done* logic. The TCSM (Figure 13) is modeled after the IEEE 1149.1 state machine. Effectively, we have merged the command register (specified by the standard) into the TCSM to create a set of hard-coded test modes. As with the IEEE 1149.1 state machine, a single input bit, the *Test Mode Select* (TCSM) signal is used to control the TCSM, and holding this signal low guarantees that the TCSM returns to the initial state. In Figure 13, a bolded state indicates that the TCSM loops in that state when the TMS is high. All other transitions are shown, with a solid arrow indicating the high transition and a dotted arrow indicating the low transition.

The TCSM has six encoded modes: two for test, three for configuration, and one for execution. The *logic test* (Ltest) and *memory test* (Mtest) modes are one- and two-cycle test modes, respectively. Ltest is used to test all logic paths, including the 3D interface (detailed in Section 3.4.3). Mtest is used to both test the memories and to load/unload them at the beginning/end of execution. Two cycles are required because the memories must

Table 2. List of global control signals produced by the TCSM, their functions, and the TCSM states in which they are active.

Signal	Purpose	State(s) Active
Test_en	Places entire chip in test mode	All states except Ltest_capture, Mtest_capture, EXE_run
RF_en	Allows writes to the register file	Ltest_capture, EXE_run
Mem_en	Allows writes to the IM and DM	Mtest_launch, EXE_run
PBC_hold	Freezes the contents of the pipeline bypass chain	All states except PBC_scan
CBC_hold	Freezes the contents of the core bypass chain	All states except CBC_scan
SCC_hold	Freezes the contents of the sector control chain	All states except SCC_scan

respond to the input data they receive on the first cycle. The three configuration modes are used to set the contents of the PBC, CBC, and SCC respectively. Finally, the execution mode sets the processor in functional mode and allows programs to execute. The actual execution state is sandwiched between two scan states, which allow execution to be halted and debugged by scanning temporary state out of and then back into the machine.

TCSM Control Signals The TCSM produces four critical control signals: *test_en*, *rf_en*, and *mem_en*. Test enable puts the chip into serial scan mode for test instead of parallel load mode for program execution. It is disabled only for the scan test, memory capture, and execution states. The register file and memory enable signals are used to protect the state in their respective units during scan cycles. Register file enable is enabled in scan test and execution states only, and memory enable is enabled only in memory launch and execution states. The TCSM produces a further three *hold* signals, one each for the PBC, CBC, and SCC chains, used to hold the contents of these chains once they have been programmed. All signals are summarized in Table 2. Note that these six signals (with the exception of SCC hold) that are broadcast globally and so necessitate the synchronization flip-flops discussed previously.

Functional Signals The finally component of the CTC is the reduction logic for the *barrier* and *done* signals. For *done*, the four sector signals are masked according to the SCC chains and ANDed together to produce the final, off-chip signal. Handling *barrier* is slightly more complicated. The final *barrier* signal is calculated and sent off-chip, identical to *done*. However, one last control signal, coming from TDI<0>, is ANDed into the tree before this signal is broadcast out as the barrier acknowledge; the purpose of this is to create breakpoint-style functionality.

When a program produces an erroneous result on an experimental chip like this, it is always a challenge to determine if the problem is in the hardware or the software. As such, we have maximized our program debugging capabilities. As mentioned previously, the execution state in the TCSM is both proceeded and followed by scan states. This allows us to pause the execution, read out the contents of the pipeline stages, reload these same contents back into the pipelines, and resume execution exactly where it left off. Of course, this only works if the exact cycle number of interest is known. For cases where it is not, breakpoint functionality is desired; this is where the off-chip barrier signal comes in (Figure 12, leftmost AND gate).

During normal execution, this signal is held high, and barriers resolve as quickly as possible without any outside interference. However, in debug mode, we can hold this signal low. When the program encounters the barrier, it will not resolve, and we can then read out the memory and register file contents for examination (unfortunately, the pipeline contents will be mostly lost waiting for the barrier signals to reach the CTC initially, but this is unavoidable). After the memory is read out, we set the off-chip barrier signal, and the program resumes execution. Thus, by inserting barriers at key points, we can break the program execution at any point, a very useful debugging feature. Because barriers are reported off-chip, the test system can count barriers as they occur to distinguish between breakpoints and synchronization points, allowing the latter to resolve unimpeded and maintaining full chip functionality even in debug mode.

3.4.2.4 Executing a Program

Here we describe the basic process for executing a program (assuming an all-good processor). First, we enable all sectors by setting the SCC. Then we enable all cores for scan by setting CBC appropriately. Third, using PBC, we enable only pipeline stages one and three. Then we loop through the memory load/test branch of the TCSM a few thousand times to fill the IMEM and DMEM. Now that the program is loaded, we enter the *EXE_init* state. In this state, we scan all zeros into the chip, a state architecturally defined to be safe.[2] We also use this state to ramp the clock up from test frequency to core frequency, if desired.

Next, we enter one final preparation state, which ensures the initial PC is correctly read[3]. Finally, we execute the program. Upon receiving the *done* signal, we return to the memory load/test branch to read out the contents of the memory and verify the output of the program—of course, setting the PBC to pipeline stage three and five only (for sending read commands and receiving read data, respectively) will speed up the read out process. This process is then repeated for each benchmark and data set.

3.4.3 Testing 3D-MAPS

The 3D-MAPS chip has been fabricated, packaged, and mounted to a test system at the Georgia Institute of Technology, School of Electrical and Computer Engineering. We have found significant success applying test patterns with the described test architecture. Numerous bugs have been removed from both the C++ model and the test system RTL. We have even discovered and resolved a couple of discrepancies between the 3D-MAPS RTL and the actual chip. These discrepancies were quite unexpected, since the chip was compiled and implemented via CAD tools directly from the RTL description.

So far, testing has shown that 3D-MAPS has been fabricated exactly as described in our GDSII files; no manufacturing bugs have been found. One design bug has been discovered. In our design methodology, we adopted an active-high standard for enable signals (i.e. a

[2] "Safe" means the program state (i.e. IMEM, DMEM, and register file contents) will not change.

[3] In normal operation, the PC loads either the previous PC or the branch target incremented by eight. The extra cycle is required to avoid that plus-eight calculation.

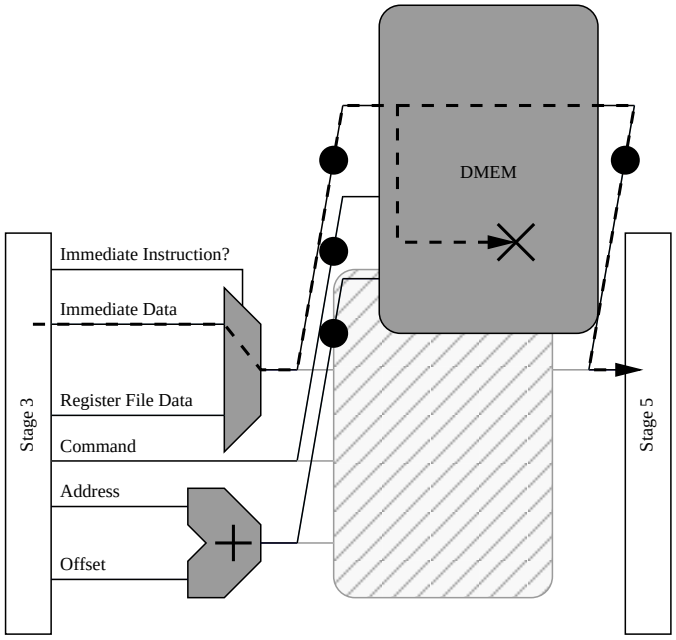


Figure 14. A schematic of the path utilized to verify the 3D interface. A shadow of the DMEM is shown on the bottom tier to more clearly show its functional connections to the rest of the processor.

unit is active when its enable signal is a logical one). The memory compiler used to produce the IMEM and DMEM however used active-low enables. In most cases, the conversion from active-high to active-low was properly made. However, one case was missed. It occurred in the logic that controls for a writing a character or word to the DMEM. This has proven to be a very minor bug and requires only that we play some tricks to fill up the DMEM with data. Additionally, this bug discovery informed the design of version two of 3D-MAPS (which is currently with the fab) by helping us identify and fix a related design flaw in that chip. Overall, it was a very beneficial experience.

3.4.3.1 Testing the 3D Interconnect

Here we describe the process required to test the 3D interconnect. Figure 14 shows in detail the 3D path. A memory instruction is launched from pipeline stage three. The instruction (composed of data, address, and the memory command) passes through some logic before traversing the 3D, microbump bus and arriving at the memory. The memory unit simultaneously executes the instruction and immediately copies the input data to the

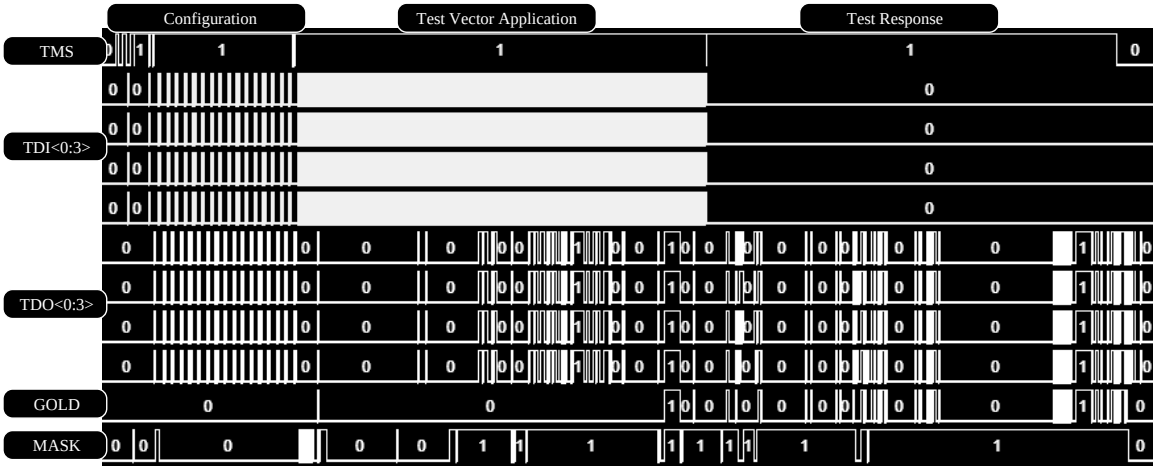


Figure 15. The output of a test of the 3D bus as captured by the logic analyzer.

output. This data traverses the 3D interface again, where it is captured by stage five.

The passing of memory data transparently to the output is quite a convenient feature. It enables a simple, quick test of the 3D bus, as traced out by the dashed arrow in Figure 14. To test the 3D bus, we launch data from stage three, allow it to propagate both into the DMEM (as marked by the *X*) and on to stage five. Without this functionality, we would be forced to execute back-to-back write-and-read pairs or insert additional DFT hardware, neither of which is ideal. This simple feature has proven quite valuable for enabling quick, direct test of the 3D bus.

Figure 15 shows an example test response when the 3D bus is exercised. From top to bottom, this screen capture shows:

1. the TMS signal
2. the four TDI signals (one per sector)
3. the four TDO responses (one per core)
4. the golden response (there is only one for this experiment because all sectors received the same input)
5. the mask (this stream identifies which bits are known versus which are don't cares)

The experiment begins with the configuration sequence for SCC, CBC, and PBC. The third visible ‘1’ on TMS marks the application of the test vector wherein the 3D bus is being activated. The fourth visible ‘1’ on TMS indicates that scan out of the test response is occurring. A comparison between the TDO streams and the golden response reveals that 3D-MAPS passed the test. There are a few discernible discrepancies, but they match up perfectly with lows in the mask stream and so are not relevant to the test of the 3D bus.

3.4.3.2 Other Experiments

We have sampled many other paths within the chip beyond the 3D bus. So far no manufacturing bugs have been discovered. While this is consistent with a mature process like 130nm, it is surprising how robust the 3D process appears to be. Most importantly, the configuration chains have been fully vetted with a number of fully random test patterns that pushes their functionality to the limits. Other paths such as the DMEM, IMEM, register file, ALUs, and bypass networks have only received limited testing. We expect to fully validate the manufacturing quality of these paths as well as our test capabilities improve.

We have also collected some initial results for power consumption. These results suggest that the simulated power numbers are quite reliable (approximately 20% error). They also suggest that the chip is operating stably at 277MHz. More definitive frequency and power results must wait on further development of the test system, as described previously.

3.5 Summary

In this chapter, we presented a new DFT architecture for enabling pre-bond test of 3D die. This architecture is based on the generic test wrapper design, which has already been successfully applied to board-level and SOC test. In this case, we treat each tier as a separate test module. Each tier test wrapper is complete with an LTC and boundary registers. These simple test features suffice for most designs; specifically, this design has been used to great effect in the 3D-MAPS test chip.

We also presented a few tricks for maximizing the benefit of implementing a product in 3D while maintaining pre-bond testability and for minimizing the cost of pre-bond testability by amortizing the cost of test resources across several different use cases. Our hybrid signal distribution network creates a minimum amount of active wiring post-bond while maintaining complete functionality pre-bond. Our pre-bond probe pad reuse scheme utilizes the pads in a new way post-bond to maximize the benefit of these costly structures. Taken together, this work establishes a strong foundation for designing fully testable 3D integrated processor systems.

CHAPTER 4

3D CIRCUIT DESIGN FOR PRE-BOND TEST

The previous chapter focused on archecture-level partitionings of 3D designs, wherein the units making up the chip architecture are spread across the tiers but each individual is whole and functional. While this is certainly a powerful design option, even more effective 3D designs are possible if we start to partition the units themselves across multiple tiers. This so-called circuit-level partitioning offers the greatest performance benefits but also poses the toughest challenges to designers. In this chapter, we take a look at a couple circuit-partitioned 3D designs and tackle the problem of testing their component pieces pre-bond.

4.1 3D Circuit Design and Test

Previous work in 3D design has examined different partitioning schemes for key functional units in high-performance microprocessors. These units include caches [66], instruction schedulers [67], arithmetic units [68], and register files [69]. Some of these—the cache designs in particular—involve what is best described as sub-block partitioning. These designs are easily testable using the wrapper-based test strategy discussed in the previous chapter. Others, most notably the port-split register file design, are partitioned at a very fine granularity and seem completely untestable by known techniques.

To cover this range of partitioning options, two designs are selected as representative cases. These are the bit-partitioned Kogge-Stone adder and the port-partitioned register file. The Kogge-Stone adder represents the easiest of the circuit-partitioned cases, using only a few internal 3D vias and mostly resembling an architecture-partitioned design (i.e. most functionality is still intact pre-bond). The port-split register file, on the other hand, makes extensive use of internal 3D vias and heavily divides functionality across tiers, representing a unique and difficult pre-bond test challenge. These two functional units, an adder and

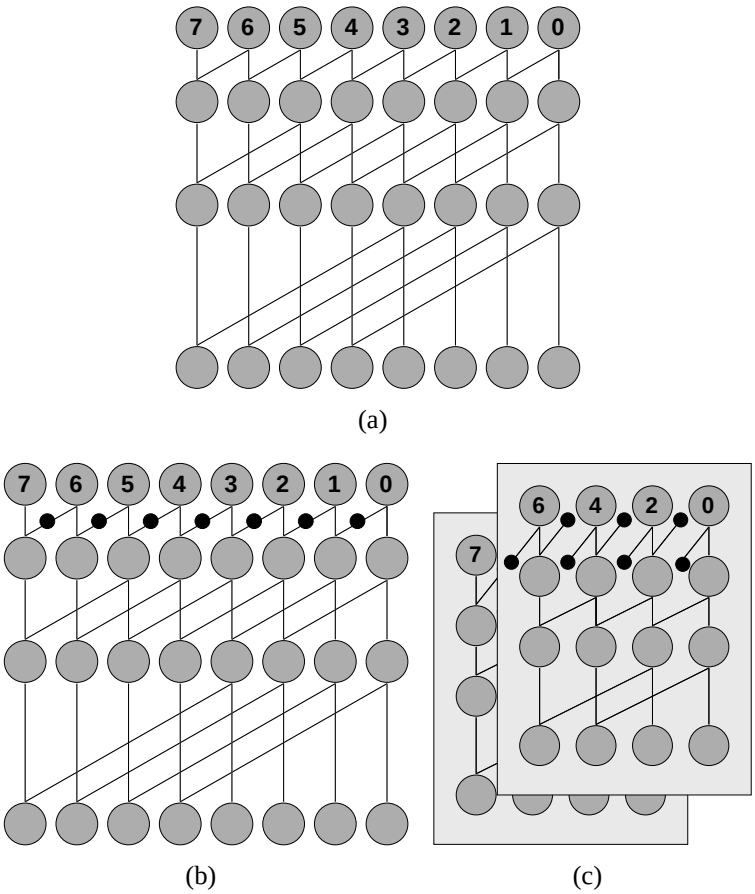


Figure 16. An 8-bit Kogge-Stone adder. (a) shows the planar implementation with its massive wiring area. (b) shows the placement of the 3D vias in the 3D design. (c) shows the true 3D design with the significant wiring reduction.

SRAM memory array, also represent the most commonly seen components inside a microprocessor The particulars of each 3D design and the necessary test strategy are discussed below.

4.1.1 Kogge-Stone Adder

The planar and 3D designs of an eight-bit adder are shown in Figure 16. A Kogge-Stone adder makes heavy use of prefix units to minimize the fanout of each unit and increase addition speed. As shown, prefix values are shifted left after each stage by an exponentially increasing distance to produce the carry values. As the bit count increase to 32, 64, and 128 bits, the wiring costs explode. To alleviate this problem, the 3D design proposes a modulus

partitioning of the original operand bits. Figure 16(b) shows a modulus two (i.e., odd and even) partitioning. In the first level of logic, the even bits and odd bits are exchanged across 3D vias. In the last logic level, the generated carries must be shuffled because they are generated on the wrong tier from which they are used. In all other logic levels, the even and odd halves of the adder do not communicate. While the planar implementation had to wire these non-communicative blocks side-by-side, the 3D partitioning enables the independent wiring to get out of each others' way, greatly reducing wiring area. Note that the wiring complexity of the 3D implementation resembles that of a planar 4-bit adder, a significant improvement over the 8-bit planar adder. So modulus two bit-partitioning has the effect of replacing the last, most-complex tract of wiring with a via tract (with wiring complexity equal to the first, simplest wiring tract), significantly increasing addition speed while simultaneously cutting power consumption.

Though only a modulus two partitioning is shown, higher moduli can be used in taller stacks. For example, with four tiers, each group of four bits could be partitioned across the stack. This would replace the two last, most complex wiring tracts with two via tracts of complexity equal only to the first two wiring tracts. Thus the design is very extensible to higher tier counts.

4.1.2 Testing the 3D Kogge-Stone Adder

The 3D Kogge-Stone adder has 3D vias only in the first few and last logic levels. Thus, these vias are easily accessible from outside the adder as control points. To test the adder pre-bond, we simply add scan registers at the edge to provide test values on these nets. This enables full structural test of each half of the adder pre-bond.

Because test cost (i.e., number of applied patterns) in general grows superlinearly with the complexity of the circuit under test, 3D designs naturally reduce total test time. That is, the number of patterns required to test each tier independently pre-bond and then test the connecting 3D vias post-bond is much less than the number of patterns required to test the

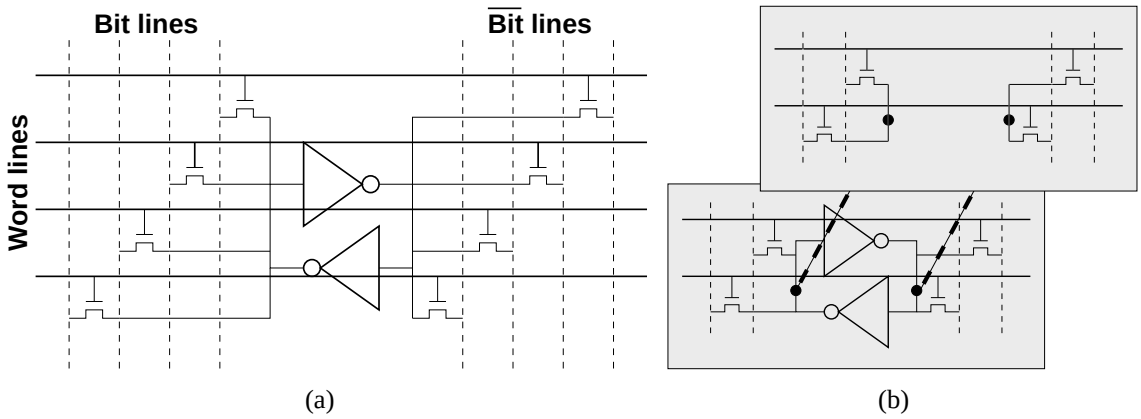


Figure 17. A 4-port SRAM cell. This cell is laid out in an array to form a 4-port register file. (a) shows the planar implementation with its massive wiring area. (b) shows the equivalent 3D design. Note that the lengths of the bitlines, wordlines, and internal nets have all be significantly reduced.

planar implementation. To be fair, the planar design could be augmented to artificially divide it into independently-testable circuits similar to the 3D division. However, this would be more costly than the 3D split because it would require insertion of multiplexors into the adder's critical path to disable functional data during test. Since there is no functional data in the 3D adder pre-bond, this extra delay can be avoided, reducing the impact of test on the normal operation of the chip. Of course, the test data must be gated post-bond, but this gating would be off the critical path and thus less of a concern.

4.1.3 Port-Split Register File

Current high-performance microprocessors require simultaneous access to many operands from the register files to maintain high instruction throughput. Typically, the requirement is two read ports and one write port per parallel instruction plus a few extra for functions such as reads for data forwarding in the load-store queue that manage memory accesses. Modern superscalar processor designs execute between two and six instructions in parallel, which would require a minimum of six ports up to twenty or more ports.

Figure 17(a) shows the planar implementation of a port-split register file. Note how the size of each bit grows quadratically with the port count, as each port requires dedicate bit- and wordlines. For a high-end, twenty-ported register file, the capacitances on the internal

nets is massive, which is not desirable as the register file is critical in determining the operating frequency. To overcome this quadratic growth, an aggressive port-partitioning design was proposed in which some of the ports (half the ports, in the case of the two-tier design shown in Figure 17(b)) are placed on other tiers. All these tiers share a single cross-coupled inverter pair, with the ports on other tiers connected back through 3D vias. In the two-tier design, this reduces the size of the internal nets by a fourth. With two tiers, this adds up to half that size of the planar design. But not only are the internal nets significantly reduced, but all the bitlines and wordlines are also cut in half, effectively reducing the wiring load of the entire register file by half. This leads to significant, simultaneous performance improvement and power reduction.

4.1.4 Testing the 3D Register File

While the benefits of port-splitting are impressive, such a design poses serious pre-bond test challenges. Most notably, before the tiers are bonded, only one tier has access to the actual storage cell. The other tiers have ports to nothing; they are functionally broken. This prevents the application of traditional memory test techniques such as Walking Ones [6] to any of these tiers. To test these tiers, a new approach is required.

Obviously, the tier with the memory cell can be tested using a classic algorithm. For the other tiers, even though the memory cell is missing and the circuits cannot be tested as a memory unit, there is still sufficient functionality left in the circuit to test it. To enable test, we split the ports in such a way as to ensure that there is at least one write port and at least one read port on each tier. If the partitioning of a particular design has only read (or only write) ports on a given tier, one port could be converted to a combination read/write port to enable pre-bond test, a minimal overhead. It is now possible to stream test data through the ports to ensure they are functioning properly. This strategy tests each write port serially. A test vector is applied to the write port. Then the address of the write port and each read port is stepped through sequentially (Figure 18). This has the effect of the write port placing a value on the internal nodes and the read ports immediately reading it. Thus, we can verify

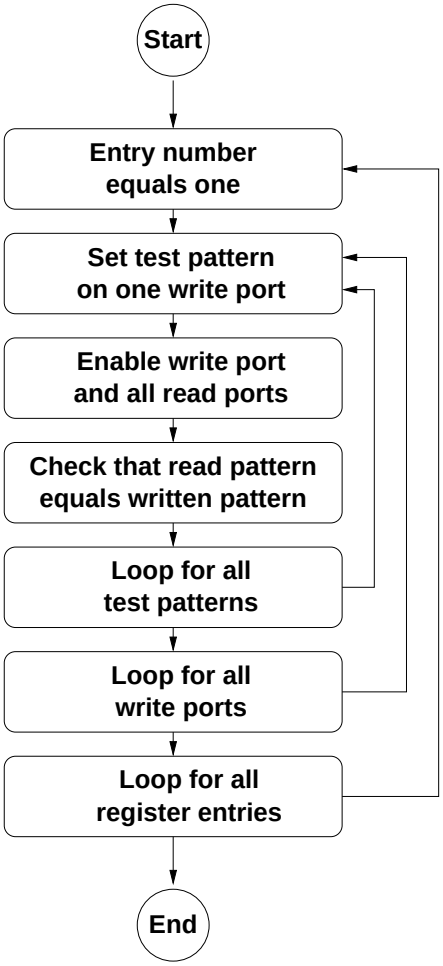
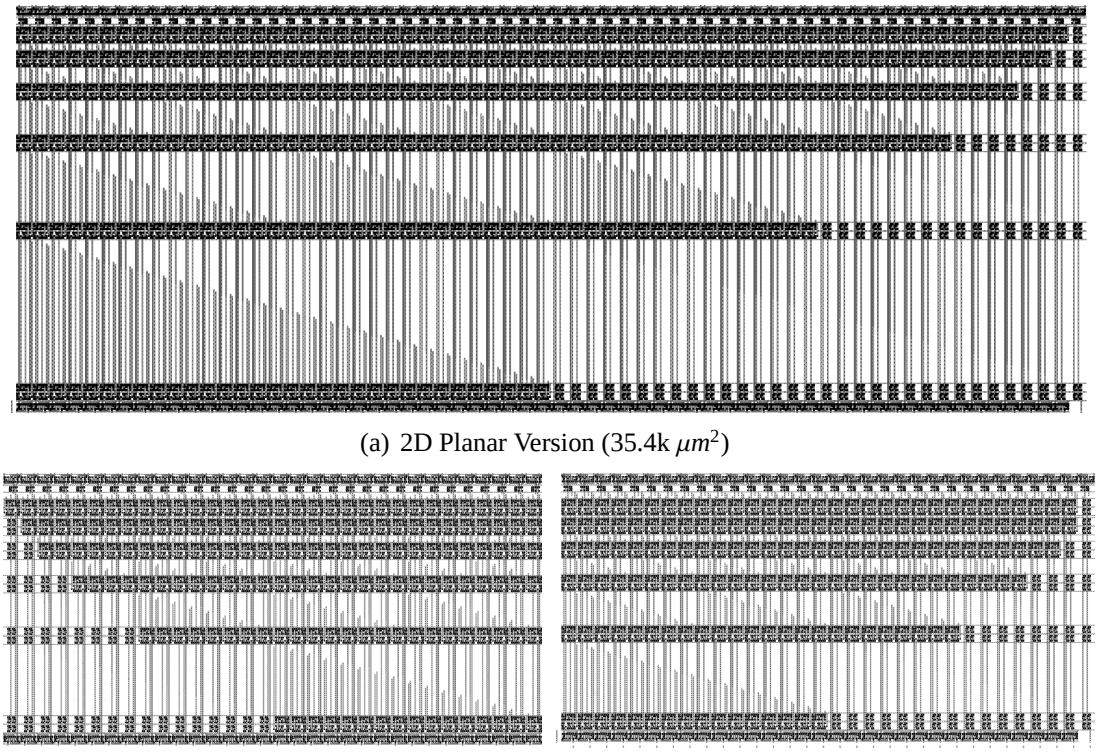


Figure 18. Flowchart of the 3D register file test algorithm.

the proper functioning of the ports by observing the initial test vectors on the read ports.

It is important to note that this strategy tests all memory components: address decoder, write hardware, bitlines and wordlines, ports, and sense amplifiers. The latter four all participate directly in passing the test data, so it is easy to see how they are tested. The address decoders, on the other hand, are tested in a slightly indirect manner. Since the write decoder and all read decoders all should be receiving the same address and producing the same one-hot register entry, a fault in one of them will activate the wrong entry and produce an error on the output. It is possible that all ports suffer from the same error and thus produce the correct output, but this would be an exceedingly rare occurrence, and such a situation could still be detected in the final memory test of the bonded stack, so this is not



(a) 2D Planar Version (35.4k μm^2)

(b) 3D Top (11.7k μm^2)

(c) 3D Bottom (11.8k μm^2)

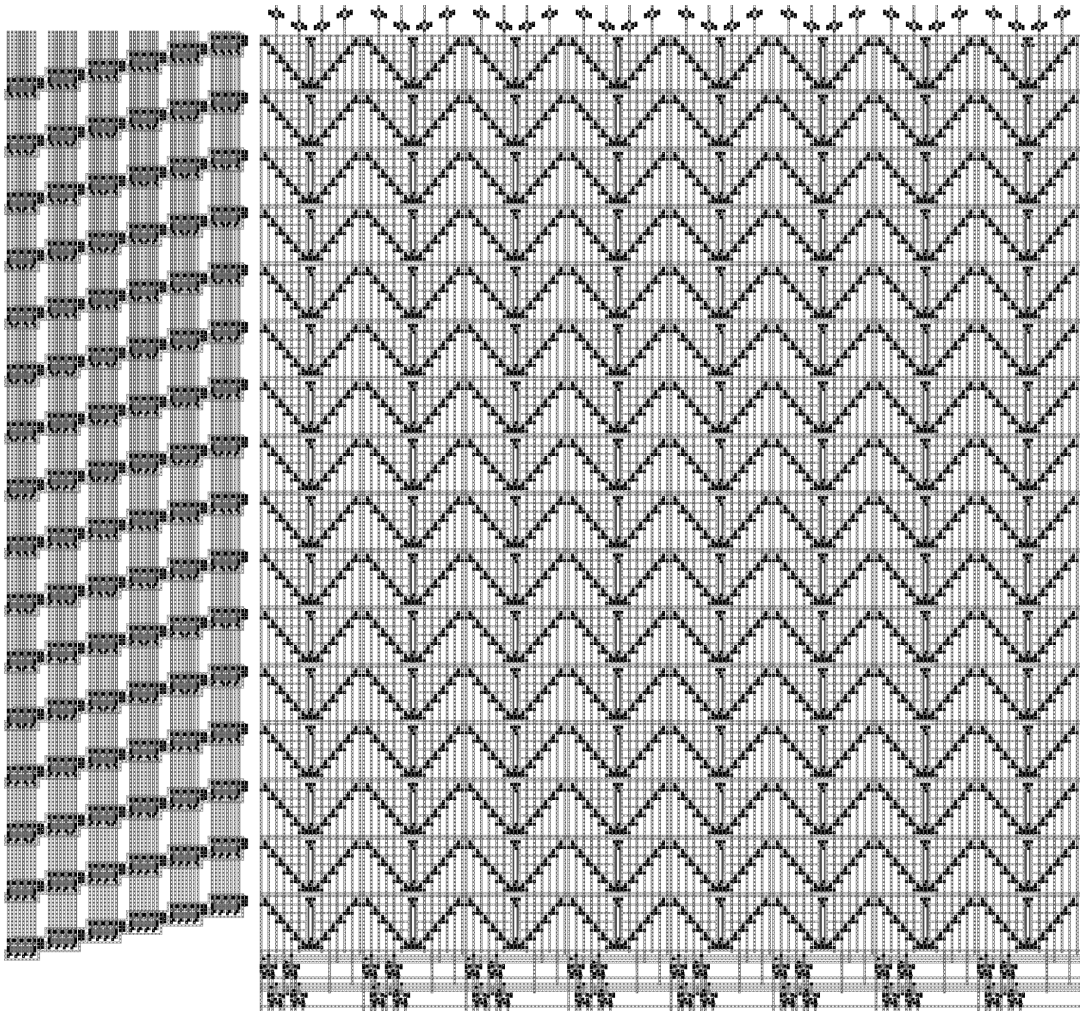
Figure 19. Layouts for a 64-bit Kogge-Stone Adder.

a concern. Thus, full test of the memory-less ports is achieved pre-bond.

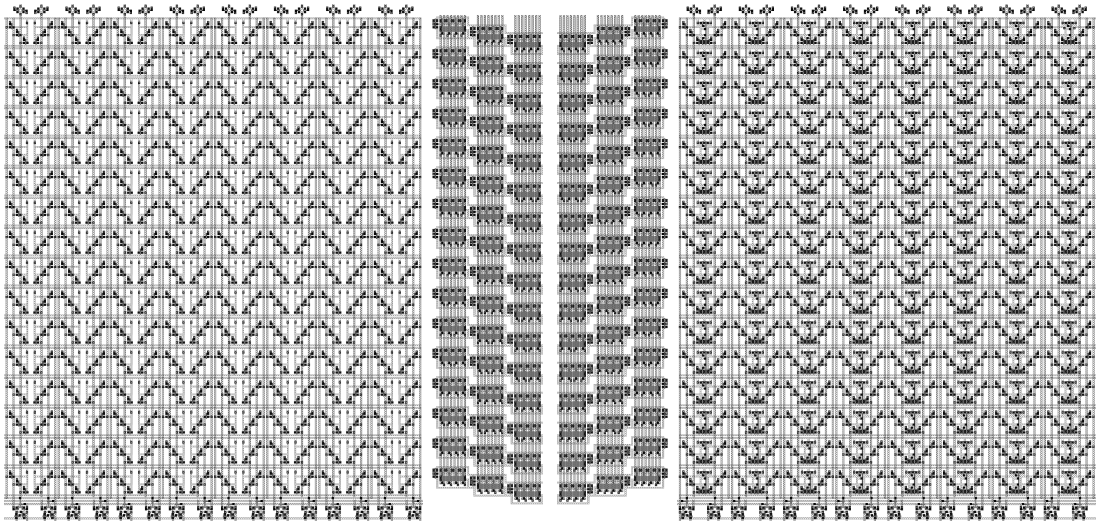
4.2 Experiments

4.2.1 Power and Performance

To evaluate our test strategy on these two circuits, planar and 3D versions were implemented in 3DMagic [24], an extension to the open-source Magic VLSI tool [1], that enables the creation of 3D layouts. Both implementations were partitioned across two tiers. Our register file implementation is a 6-port (four read and two write ports), 8-bit, 16-entry design appropriate for a two-instruction-wide processor (Figure 20). The layout consists of four main components. First and most important is the actual SRAM cell array, which dominates each layout. Beside the SRAM array is the address decoder logic with six decoders per row, one per port. Above the array are the write drivers, two per column for the write ports. Last are the sense amplifiers below the array, four per column for the read



(a) 2D Planar Version (20.3k μm^2)



(b) 3D Top (6.24k μm^2) (c) 3D Bottom (6.24k μm^2)

Figure 20. Planar and 3D layout for a 6-port 16x8b register file. Despite the large area difference, these two designs have equal storage capacity.

ports. It is important to note that, within the SRAM array, each dark spot is a transistor. Because multiported register files are wire-dominated, the transistor density is very low and a lot of silicon is going to waste.

The 3D implementation, in constrast has a much higher transistor density and makes much better use of the available silicon. In this implementation, two read ports and one write port were placed on each tier. As reported in Table 3, the 3D implementation achieves the same memory capacity as the standard register file but at only 61% the silicon cost. Futhermore, the 3D footprint is over three times smaller than the planar footprint, which may be a crucial benefit since most chips are limited in size by packaging restrictions.

Our Kogge-Stone implementation is a full 64 bits as shown in Figure 19. To compute a 64-bit sum, the Kogge-Stone adder requires eight levels of logic. The first level, located at the top of the layout, computes the generate and propagate signals. The next six levels incrementally gather the *p and* g signals to produce a carry for each bit. As Figure 20(a) demonstrates, this process is completely dominated by the wires shuffling the *p and* g signals around. The final logic level, located at the bottom of the layout, produces a summation from the carry bits.

In our 3D implementation, 3D vias are required between the first and second logic levels and between the seventh and eighth logic levels. This first set of vias is the key to the implementation's efficiency, as it greatly reduces the wiring congestion. The second via array is required because the carries are generated on the wrong tier and must be passed to their proper tier. The area overhead of these vias is easily hidden in the logic levels that use them and thus do not affect the overall area. As with the register file, the 3D adder significantly reduces the area and footprint compared with the standard planar implementation (Table 3).

We were able to extract the Kogge-Stone adder from Magic to produce a generic, lambda-based circuit description that can then be used with any transistor generation description. We exported the extracted circuits to HSPICE and simulated them using a 130nm,

Table 3. This table list the area and footprint requirements for each design. The percentage listed is the size difference between the complete, bonded 3D adder and its planar counterpart.

Design	Area		Footprint	
	(μm^2)	Diff	(μm^2)	Diff
2D Adder	35.4k		35.4k	
3D Adder - Top	11.7k			
3D Adder - Bottom	11.8k			
3D Adder	23.5k	66%	11.8k	33%
2D Register File	20.3k		20.3k	
3D Register File - Top	6.24k			
3D Register File - Bottom	6.24k			
3D Register File	12.5k	61%	6.24k	31%

Table 4. This table gives the power and performance numbers for the two adder implementations.

Design	Cycle Time		Power	
	(ns)	Diff	(mW)	Diff
2D Adder	7.46		26.1	
3D Adder	6.08	82%	22.6	87%

level 49 transistor model. The power and performance numbers for the Kogge-Stone adder are presented in Table 4. The 3D adder obtains, simultaneously, a 18% cycle time and 13% power reduction. This means that a 3D adder can run at a significantly higher frequency than a planar version for equal power consumption, or it can run at equal speed for a nice power savings, depending on the needs of the design. This work verifies the power and performance results of the previous work [68] which were based on critical path estimations of the circuits.

4.2.2 Test Cost and Coverage

To evaluate the test cost and coverage for the Kogge-Stone adder, we used the Mentor Graphics tool set [52]. First, gate-level structural Verilog models of both the 2D and 3D implementations were produced and verified in ModelSim. For the 3D case, we produced three model files: one file describing the bottom tier, one file describing the top tier, and one file describing the 3D via connections. This division of the model ensured an accurate

Table 5. Listed are the pattern counts required to test each part of the design. These patterns were obtained from deterministic ATPG.

Design	Pattern Count
2D Adder	313
3D Adder - Top	146
3D Adder - Bottom	145
3D Adder - Vias	10
Total	301

description of the model was available for both pre- and post-bond test simulation.

The actual test simulation was produced using FlexTest. This tool provided a list of faults, a set of test vectors, and the fault coverage achieved. In order to achieve a fair comparison between the planar and 3D cases, we ran three fault simulations for the 3D implementations. The first two targeted all faults within the two independent tier models, simulating pre-bond test. The last simulation targeted faults on the 3D via nets between the two tiers, simulating a post-bond test verifying that the two tiers were successfully bonded. Summing the cost of these three tests estimates the total cost of testing the 3D design fairly,

The test simulation results are reported in Table 5. In confirmation of our earlier hypothesis, the combination of testing the top tier, bottom tier, and interconnecting 3D vias required less patterns than testing the singular planar design. More importantly, note that the top and bottom tiers, being independent DUTs during tier test, may be tested in parallel. This means that while the 3D design uses only 0.4% fewer patterns, it can be tested in just 156 cycles or in 49.8% of the time required for the 2D test.

The register file, being a RAM structure, requires a test methodology very different from the adder. Because this register file is a relatively small structure, we can reasonably apply a fairly complex test pattern. For comparison, we use Suk and Reddy's Test B [76], adapted to multiported structures. The single-ported algorithm requires $16n$ accesses, where n is the number of bits (128 for our register file). To accomodate multiple ports, we multiple by $max(readports, writeports)$. This comes out to 12.3k accesses to test

the planar register file.

For our 3D register file, we apply Test B to the bottom tier (containing the state logic), requiring 6144 accesses. Implementing the algorithm described in Figure 18 requires $2n$ accesses, another 256 patterns. Of course, once the tiers are bonded, we must test the 3D via connections, which requires $4n$ or 512 patterns. Thus, in total, testing the 3D version of this register file requires just 6912 accesses, which is far superior to testing the planar design. In this case, simplifying the circuit with partitioning has greatly improved the test situation.

4.3 Summary

In this chapter we have investigated test strategies for circuit-partitioned 3D designs, in which a functional unit can be partitioned into incomplete circuits across different tiers. Our techniques present standard scan registers that can be integrated into the tier scan chains, allowing the ATE to (in the standard scan case) directly test the circuit or (in the PRPG/MISR case) initialize the registers for BIST. To demonstrate our methodology, we performed two case studies using a prefixed parallel adder and a register file. In the case of the bit-split 3D Kogge-Stone adder, pre-bond test involved a simple extension to scan-based test. The port-split 3D register file, was much more difficult, requiring a new test strategy to enable pre-bond test. Our full layout implementations confirmed the power and performance improvement estimates reported by previous work, and our fault simulations based on detailed Verilog models demonstrated high fault coverage at reduced cost compared to equivalent planar designs. We have shown that even the most difficult 3D partitioning schemes can be tested pre-bond, ensuring the viability of many-tier chip stacks.

CHAPTER 5

3D TEST WRAPPERS

Chapter 3 presented a general tier test wrapper for enabling pre-bond test in 3D systems, while the previous chapter presented some ad hoc methodologies for testing circuit-partitioned 3D designs. Unfortunately, these solutions hinge on the critical assumption that the entire 3D system is known to the test architect. This is not generally true, as ICs increasing contain IP blocks not owned by the system integrator. This requires a more advanced 3D test architecture standard for allowing black-box testing.

The natural solution is to extend the design standards of IEEE 1149.1 and 1500 to 3D, and this work is well under way. Wu et al. [84] designed test time optimized TAM architectures for 3D SOCs under 3D via count and TAM bandwidth constraints. This work concerned itself just with planar cores in a final stack test mode. Noia et al. [58] designed test time optimized wrapper chains for 3D cores with 3D internal scan chains under a 3D via count constraint. This work also focused on final stack test without considering the implications of pre-bond test.

Jiang et al. [32] designed 3D TAM architectures that optimized the total test time—pre-bond and post-bond testing. However, they considered only planar cores with a single fixed wrapper for both test modes. In a follow-up work [33], they designed separate pre-bond and post-bond TAMs and developed a methodology for sharing routing resources between these TAMs. Consideration was still limited to a singular wrapper for each core. Lo et al. [45] developed a 3D TAM architecture called TACS-3D, a daisy-chaining scheme for 3D SOC test. This work treats 3D vias as I/O connections and uses standard boundary scan designs to test these connections. This work also limits its considerations to planar cores.

Marinissen et al. [50] proposed an extension to the 1500 standard for 3D ICs; this extension is a die-level test wrapper that includes probe pads on every tier for pre-bond test, *test elevators* for accessing probe-inaccessible tiers in partial stack and final stack test,

and a hierarchal *wrapper instruction register* for test control. This work also only considers planar cores. More interesting though, it allows for the number of probe pads used in pre-bond test to differ from the number of test elevators used in partial and final stack test. This work does not address how the die wrapper handles these two different TAM bus widths. In a follow-up work [48], this work is extended to demonstrate that an 1149.1 style embedded wrapper may also be used for the die-level wrapper; the limitations of the previous work remain.

Here, we propose a new test wrapper design algorithm for 3D IP blocks.

5.1 Problem Definition

In wrapper-based DFT, a *core under test* (CUT) is assigned some number of parallel test channels for loading and reading test patterns. However, 3D CUTs add a new twist: the number of test channels available to the CUT may differ in the pre-bond and post-bond test modes. If the test access width is to change, the wrapper must include the flexibility to adapt to the different widths.

5.1.1 Motivating Example

Figure 21 illustrates the challenge and opportunity of wrapper design for 3D IP cores. In this example, the core consists of two tiers. Each tier consists of two scan chains which must be ordered in the wrapper. Assume that the pre-bond test width for each tier is a single bit. The two scan chains on each layer are necessarily stitched together by a wire in the test wrapper to form a single wrapper chain (Figure 21(a)). Herein lies the optimization opportunity: it would be desirable to reuse that stitching wire in the post-bond wrapper in order to reduce the total wrapper wire length.

Figure 21(b) and Figure 21(c) illustrate this opportunity. Both post-bond orderings are based on a post-bond test access width of two bits. Figure 21(b) stitches the long scan chain on each tier to the short chain on the other tier; this solution fails to reuse the pre-bond

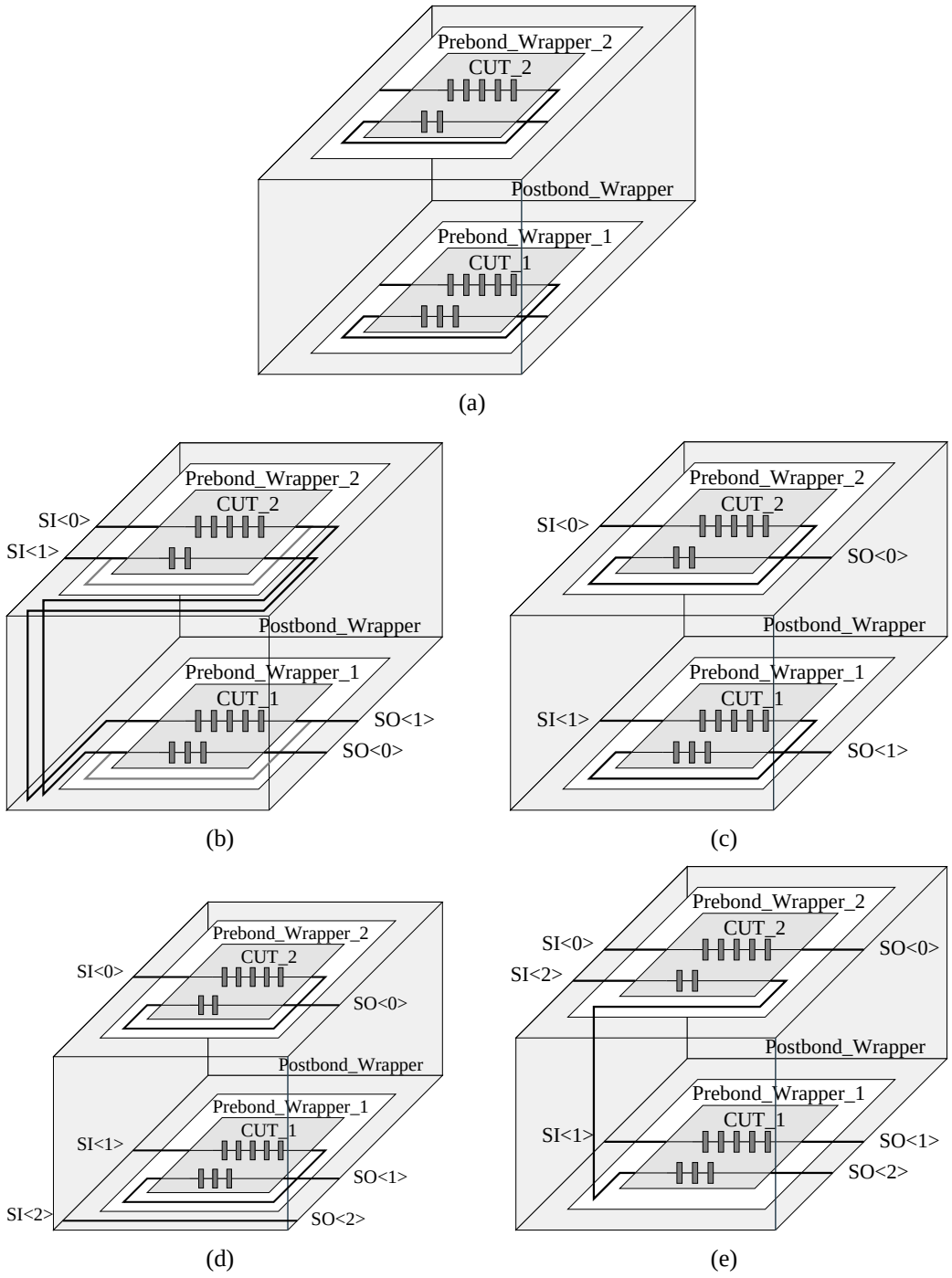


Figure 21. An example $3DP_W$ problem with four solutions (for TAM widths of two and three). (a) shows the pre-bond wrapper chain assignments. (c) and (e) are desired solutions while (b) and (d) are suboptimal.

stitching and necessitates the use of two additional 3D vias dedicated to test. Figure 21(c)[1] on the other hand does reuse this stitching. Both solutions are minimum test time for the given TAM width, so both solutions would be considered optimal solutions to the post-bond ordering problem in that sense. However, the second ordering is clearly superior when the additional cost of wire length is considered.

Figure 21(d) and Figure 21(e) motivate the weighting of the two design goals, test time and wire length. These solutions are based on a post-bond test access width of three bits. In Figure 21(d), wire length is given priority, and the result is that one test bit is wasted[2] and test time is increased significantly. This is suboptimal because test time is one of the most significant components of product cost. In Figure 21(e), test time is given priority. This solution requires some additional wires but significantly improves test time, a much better solution. Thus wire length is used as a secondary constraint behind test time for determining an optimal 3D test wrapper.

5.1.2 Problem Formulation

We define the 3D IP wrapper design problem $3DP_W$ as follows. Given a 3D IP core test description (number of I/Os, number of scan chains, length of the scan chains, and a 3D partitioning of these resources), the set of pre-bond test access bus widths, and the post-bond test access bus width, determine the optimal ordering of the I/Os and scan chains into both pre-bond and post-bond wrapper chains such that the test time is minimized and that the wirelength is minimized subject to the test time.

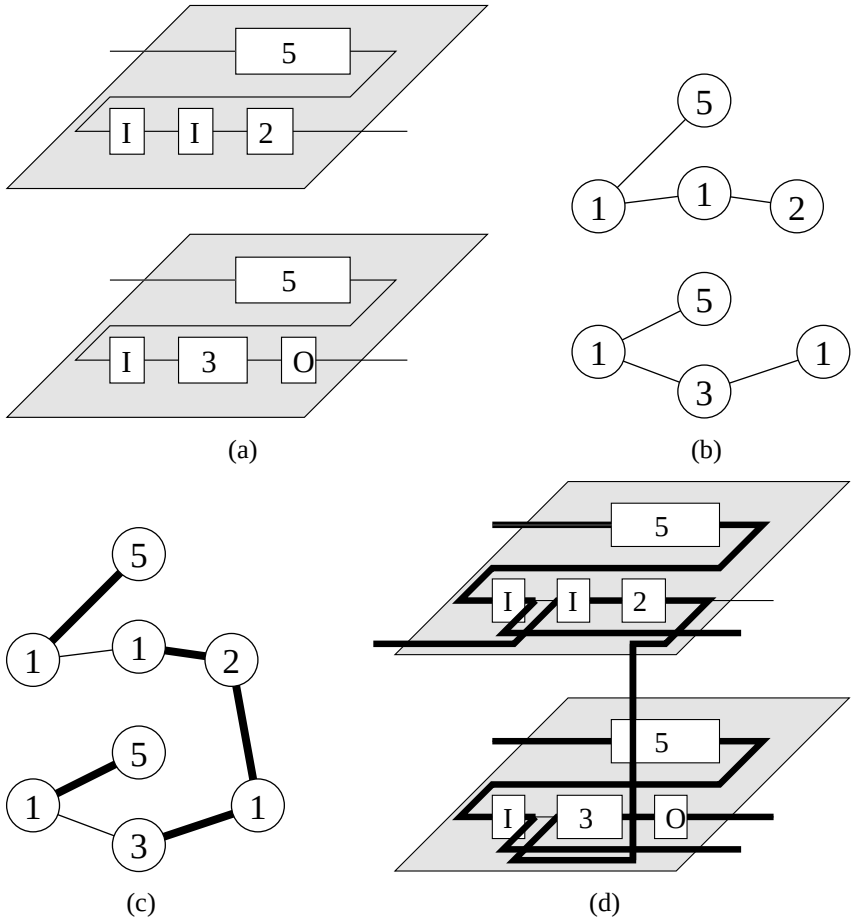


Figure 22. KL partitioning for post-bond wrapper design. (a) shows the pre-bond wrapper solutions produced by BFD. (b) is the graph representation of those solutions. (c) is the post-bond wrapper solution generated by KL partitioning. (d) is the final solution after scan element ordering. Shown is a high-quality solution in which almost all pre-bond stitching is reused post-bond.

5.2 Wrapper Design Algorithm

5.2.1 Pre-bond Wrappers First

To design the 3D test wrappers, we use a three-step algorithm. We first describe its operation assuming the pre-bond wrappers are designed first and the post-bond wrapper second. We then discuss reversing this ordering at the end of this section. Briefly, the first step applies the *Best Fit Decreasing* (BFD) heuristic to design the pre-bond test wrappers for

[1]SI and SO pins locations differ for clarity of the figure. In practice, these pin locations would be fixed as part of the contract between the wrapper designer and the TAM architect.

[2]The unused test bit could potentially be reassigned to another TAM as part of a wrapper-TAM co-optimization problem. This problem has been studied previously in [31]; the solution proposed there remains applicable in the 3D SOC case.

each tier. Step two uses the *Kernighan-Lin Partitioning* (KL) heuristic to determine optimal wrapper chain assignments for the post-bond wrapper. Finally, step three orders the post-bond wrapper chains to maximally reuse the pre-bond stitches.

5.2.1.1 Best Fit Decreasing

Designing each pre-bond wrapper is nearly identical to designing a planar wrapper. The only difference is the 3D vias. Here, the product engineer has a choice. He can treat the 3D vias as pre-bond-untestable internal nets as in [40, 41] in which case they do not affect the impact the wrapper design. Alternatively, he can treat them as inter-core communications pins as in [50] in which case they are treated like any other I/O connection in the wrapper by being assigned a boundary cell. This choice can be made on a via-by-via case, designating each as is appropriate.

To solve the pre-bond wrapper design problem then, we use the BFD heuristic [31]. We choose this heuristic because it produces a test-time-optimal pre-bond wrapper chain assignment while minimizing the use of TAM resources. BFD produces a set of wrapper chains composed of *scan elements* (the internal scan chains and I/O cells) and stitching wires. The goal of step two is to reuse these stitching wires to the greatest extent possible.

5.2.1.2 Kernighan-Lin Partitioning

To design the post-bond wrapper, we treat it as a partitioning problem. The input is a set of disjoint subgraphs. The subgraphs represent all the wrapper chains from all the tiers in the pre-bond wrappers designed in step one. The vertexes represent the scan elements (weighted as the number of scan registers in that scan element), and the edges represent the stitching. The goal in designing the post-bond wrapper then is to determine a second set of disjoint subgraphs (representing the post-bond wrapper chains) such that

1. the maximum total weight of the vertexes in any subgraph is minimized (this equates to minimizing the post-bond test time) and

2. the greatest number of edges from the pre-bond subgraphs are reused in forming the

KL Partitioning for Wrapper Design

Input: R - graph of pre-bond wrapper assignments
K - number of post-bond wrapper chains
Output: T - graph of post-bond wrapper assignments

```
1: DesignWrapper(Graph T, Graph R, int K)
2:   if (K == 1) then
3:     T = T ∪ R; return
4:   for each(scan element se_i ∈ R)
5:     Assign se_i randomly to R_L or R_R
6:   K_L = K/2; K_R = K − K_L
7:   while (Balance is improving)
8:     while (Have legal move)
9:       R_V =GreaterWeight(R_L,R_R)
10:      if (all se ∈ R_V are locked) then
11:        No legal move; break
12:      for each (unlocked se_i ∈ R_V)
13:        Calculate balance and cut gain
14:      Move and lock se with highest balance gain
15:      Record intermediate solution and gains
16:    Search intermediate solutions for highest gain
17:    if (all gains are negative) then
18:      No longer gaining; break
19:    Accept highest gain partition
20:    Unlock all se ∈ R_L and ∈ R_R
21:  DesignWrapper(T, R_L, K_L)
22:  DesignWrapper(T, R_R, K_R)
23:  return
```

Figure 23. Pseudo-code for applying KL partitioning to the wrapper design problem.

post-bond subgraphs.

Formally, the input is an undirected graph R and a post-bond TAM bus width K. R is composed of a set of disjoint subgraphs, one subgraph per pre-bond wrapper chain per tier. Thus the number of subgraphs in R is $\sum_{i=1}^{n} k_i$, where n is the number of tiers and k_i is the number of pre-bond wrapper chains on the i-th tier. The output is an undirected graph T composed of K subgraphs representing the post-bond wrapper chain assignments.

The determination of the post-bond subgraphs is achieved through recursive application of the KL partitioning heuristic [44] (Figure 22). Psuedocode for applying KL to the 3D wrapper design problem is shown in Figure 23. The optimization goals are represented by *balance* and *cut*. Balance is the ratio of the density of the first partition to the density of the second partition, where *density* is the ratio of the total weight of the scan cells in a partition to the number of wrapper chains assigned to that partition; an *overdense* partition has too many scan cells which would lead to a long test time while an *underdense* partition can accept more scan cells without affecting test time. The ideal balance is 1, which indicates that all wrapper chains can have the same number of scan cells, a solution which offers the shortest test time. Cut is the number of edges in the post-bond subgraphs that do not overlap pre-bond edges. The ideal cut is 0, which indicates that no additional post-bonding stitching is required.

Our implementation of KL is initialized with all the scan elements from every layer grouped into a single pool (Figure 22(b)) and all the wrapper chains available for assignment[3].

Each KL step begins by assigning half of the available wrapper chains to each partition. Next the scan elements are randomly assigned to each partition while maintaining balance as best as possible. Next is the moving phase. Each unlocked scan element in the denser partition is evaluated, and the move producing the best balance is accepted (ties are broken with the cut gain). This is repeated until no unlocked scan elements are available in the denser partition. All discovered partitionings are evaluated and the one with the best balance is accepted (ties are once again broken by cut). All the scan elements are unlocked and the moving phase is repeated. This continues until no more gains in balance or cut are achieved.

The final step is recursion, where each partition is further subdivided into smaller partitions. Recursion halts when only a single wrapper chain is assigned to a given partition.

[3]The scan elements are all grouped into one large pool regardless of tier because we consider 3D vias to be free resources. This is justified by the submicron size of present day state-of-the-art 3D processing

Table 6. Two-tier circuit benchmarks.

	Two Tiers	
	Cells per Tier	Chains per Tier
ckt1	3016, 3021	6, 6
ckt2	5329, 3479	11, 7
ckt3	19,890, 19,228	40, 39
ckt4	37,359, 40,751	75, 82

The scan elements in that partition are then assigned to that wrapper chain (Figure 22(c)).

5.2.1.3 Scan Element Pairing

Once the wrapper chain assignments are complete, the final step is to order the scan elements within the chains—both in the pre-bond and the post-bond wrappers—so as to minimize the cut. This simply requires searching the list of scan elements in the post-bond wrappers, identifying all those that are assigned to the same pre-bond wrapper chains, and stitching them together accordingly (Figure 22(d)). Final ordering of these short pre-stitched chains is a simple matter that can be handled with any traditional ordering scheme [49] and so is not discussed further here.

5.2.2 Post-bond Wrapper First

It is a trivial matter to reverse the order of events. In this case, we first determine the post-bond wrapper by applying the BFD heuristic to the complete set of scan elements on all tiers. The subgraphs representing the post-bond wrapper chains are then used to guide the design of pre-bond wrapper chains. Now KL is executed for each tier with the goal of producing maximally-balanced pre-bond subgraphs that maximally overlap the given post-bond subgraphs. Finally, the wrapper chains must be ordered in a manner identical to that described previously.

Table 7. Four-tier circuit benchmarks.

	Four Tiers	
	Cells per Tier	Chains per Tier
ckt1	1507, 1512, 1510, 1508	3, 3, 3, 3
ckt2	2543, 1980, 2767, 1518	5, 4, 6, 3
ckt3	9826, 9172, 10,757, 9363	20, 18, 22, 19
ckt4	20,723, 18,135, 17,011, 22,241	41, 36, 34, 44

5.3 Experiments

5.3.1 Experimental Setup

To test our methodology, we used a custom collection of benchmark circuits taken from the OpenCores database [61] as listed in Tables 6 and 7. This benchmark suite includes a 80386 processor, a DES encryption engine, and two 256-bit pipelined multipliers of differing pipeline depths. These circuits were picked so as to cover a large range of embedded core complexities. To obtain the 3D placements of the scan chains, we first compiled the design with Design Compiler from Cadence [14]. Next we partitioned the circuits with a custom FM partitioner [23] and performed 3D placement with Encounter from Cadence. Finally, Design Compiler was again used to partition and route the scan chains.

We developed our program in C++ and executed the benchmarks on a 2.40GHz Intel Xeon processor with 1GB RAM.

5.3.2 Methodology

To evaluate our algorithm, we ran a series of tests using different design modes and different wrapper configurations. Most importantly are the three design tools:

1. All BFD (`BFD`)—the BFD heuristic is used to design both the pre-bond and the post-bond wrappers with no feedback between the two processes. This is our baseline case.

2. Pre-bond First (`PRE`)—the pre-bond first variant of our algorithm: the BFD heuristic is used to design the pre-bond wrappers. These designs are then used to drive the KL

heuristic in designing the post-bond wrapper.

3. Post-bond First (POST)—the post-bond first variant of our algorithm: the BFD heuristic is used to the the post-bond wrapper. This design then drives the KL heuristic in designing the pre-bond wrappers.

To test our algorithm under different design constraints, we vary the number of TAM bits assigned to each wrapper. For the circuits *ckt1*, *ckt2*, *ckt3*, *ckt4*, we vary the post-bond TAM width from one to twelve, eighteen, forty, and sixty respectively. For each post-bond TAM width we run three experiments:

1. Half-width (05)—the total pre-bond TAM width is half the post-bond TAM width

2. Even-width (10)—the total pre-bond TAM width is equal to the post-bond TAM width

3. Double-width (20)—the total pre-bond TAM width is double the post-bond TAM width

Here, the *total pre-bond TAM width* is the sum of the TAM widths assigned to each tier. In assigning TAM bits to each pre-bond wrapper, we divide the total TAM width as evenly as possible.

Finally, for each experiment, we design 3D wrappers for both the two-tier and four-tier implementations of each circuit.

5.3.3 Results

In this section, we consider two wrapper design metrics. The first is *critical test length* (CTL). This is the sum of the longest wrapper chain in each pre-bond wrapper and in the post-bond wrapper. Total test time is the product of the number of test patterns times the length of the longest wrapper chain, so the longest chain is proportional to the total test time. We therefore use the CTL metric because it correlates directly to the total test time

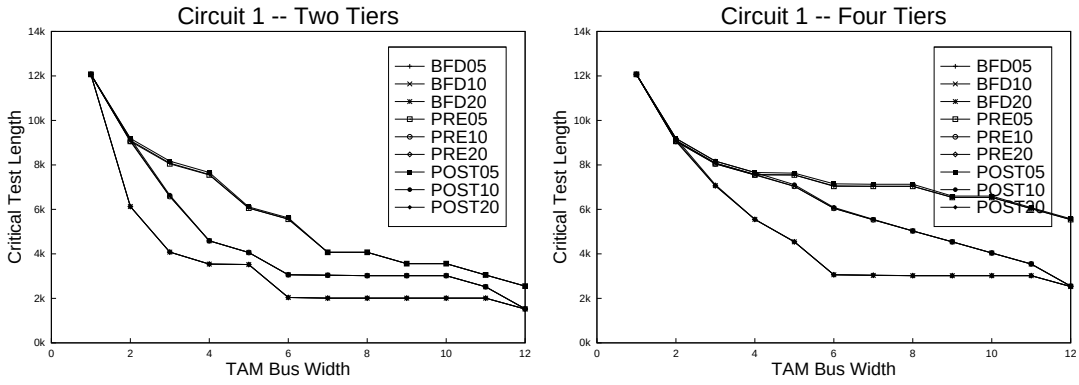


Figure 24. CTL versus post-bond TAM width for *ckt1*.

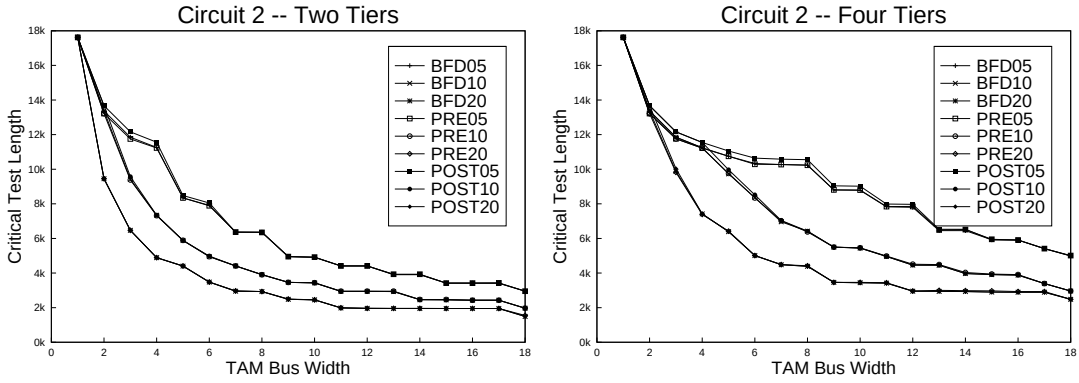


Figure 25. CTL versus post-bond TAM width for *ckt2*.

for a 3D stack (*i.e.* pre-bond test time plus final stack test time). A superior wrapper is one with a shorter CTL.

The second metric is the *cut*. This is the number of stitching wires in the pre-bond test wrappers that are *not* reused in the post-bond wrapper, basically the number of wires not reused in the post-bond wire routing. We choose this metric because fewer reused wires correlates to greater wrapper wirelength and routing congestion. A superior wrapper is one with a smaller *cut*.

The CTL results are shown in Figures 24, 25, 26, and 27. In these results we can see some very clear general trends. First, the CTL drops continuously with some plateauing

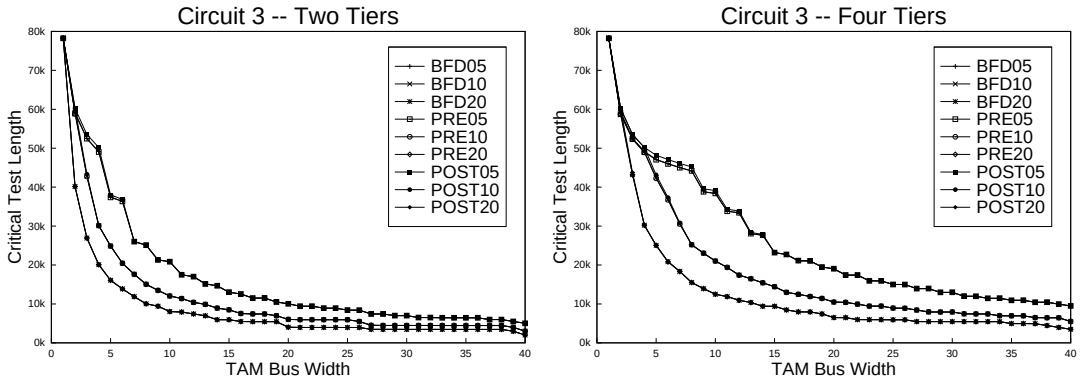


Figure 26. CTL versus post-bond TAM width for *ckt3*.

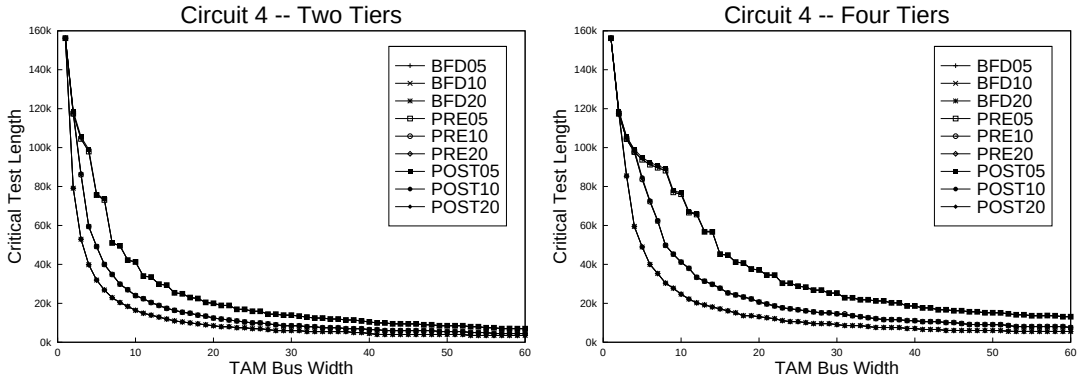


Figure 27. CTL versus post-bond TAM width for *ckt4*.

(beginning at local Pareto optimal points). These plateaus are local minima where slight increases in the TAM resource allocation are not sufficient to break up the longest chain and improve the CTL. Second, `05` wrappers have the longest CTLs with `10` wrappers doing better and with `20` wrappers better still. This is simply because those designs have more pre-bond TAM bits and so shorter wrapper chains.

Finally, the four-tier designs have highers CTLs than their equivalent two-tier designs at larger bus widths. This is an artifact of the way the CTL metric is defined, not a true result. Compared to the four-tier designs, the two-tier pre-bond wrappers have both twice

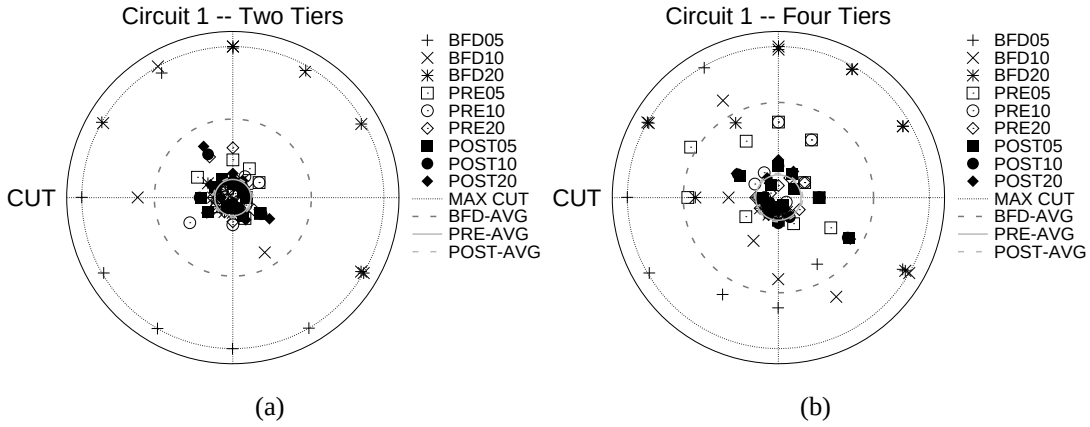
Table 8. Average percentage of pre-bond stitches for each experiment and for each method overall.

	Tiers	ckt1	ckt2	ckt3	ckt4	ALL
BFD	2	52%	15%	23%	16%	27%
	4	63%	53%	35%	31%	
PRE	2	12%	5.8%	5.0%	6.7%	6.6%
	4	15%	7.6%	5.0%	7.4%	
POST	2	13%	4.0%	7.6%	8.8%	8.4%
	4	16%	6.1%	7.3%	11%	

as many scan chains to assign and twice as many wrapper chains into which to make assignments. The two- and four-tier pre-bond wrappers therefore have approximately equal longest wrapper chains. When calculating CTL, this longest chain gets added in four times for the four-tier designs, but only twice for the two-tier designs, causing the artificial inflation in the CTL for the four-tier designs. In practice, the two- and four-tier designs would have the same test time when ATE resources are considered.

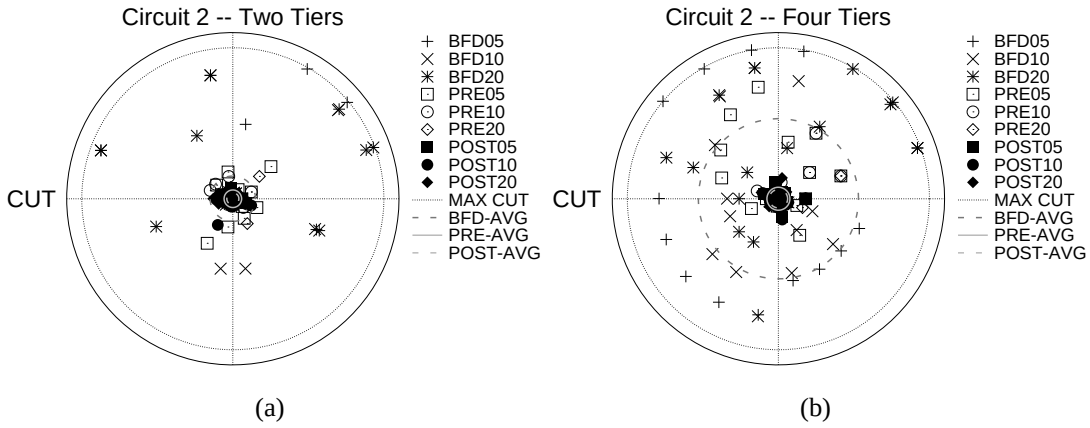
More important than these trends is the near-exact match of the CTL curves for PRE- and POST-designed wrappers to the BFD curves. Since the BFD algorithm is producing minimum test time wrappers, this close fit demonstrates that our KL-based algorithm successfully minimizes the total test time as well. On average, PRE and POST CTLs are just 0.06% and 0.32% longer than BFD respectively. In the worst case, they are still just 4.2% and 3.0% longer respectively, and a product engineer could avoid these worse cases by simply running our algorithm several times on the same input set, utilizing the random initial partitions in the KL step to find a best-test-time solution.

The results for *cut* are shown in Figures 28, 29, 30, and 31. In these polar graphs, the angle represents the post-bond TAM width (normalized to the $[0, 2\pi]$ range), and the radius represents the *cut*. The greater the distance from the center, the higher the *cut* and so the worse the solution. Also shown are four rings, indicating the max possible *cut* and the averages for BFD, PRE, and POST; these averages are also listed in Table 8.



(a) (b)

Figure 28. Polar plots of *cut* versus post-bond TAM width for *ckt1*. The four rings highlight the averages and max *cut*.



(a) (b)

Figure 29. Polar plots of *cut* versus post-bond TAM width for *ckt2*. The four rings highlight the averages and max *cut*.

In general, the results for BFD (plotted with the asterisk-style icons) are chaotic. Sometimes the *cut* is very low, and sometimes it is very high, but in general the results do not cluster at any one radius. Since there is no communication between the pre-bond and post-bond design steps, this result is expected. Sometimes the design tool gets lucky and groups the same scan chains together in both wrappers; sometimes it splits them up. BFD averages a 27% cut of the pre-bond stitching; it simply cannot reliably produce a low-*cut* design.

In significant contrast, both the PRE (represented by the open icons) and POST (represented by the filled icons) design tools consistently produce low-*cut* 3D wrappers. This result is highlighted by the tight clustering of these data points in the middle of the plots.

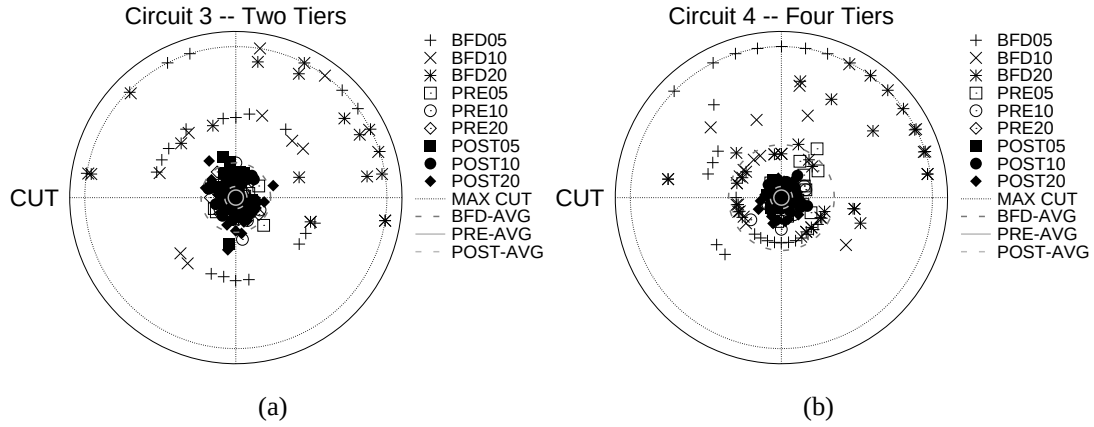


Figure 30. Polar plots of *cut* versus post-bond TAM width for *ckt3*. The four rings highlight the averages and max *cut*.

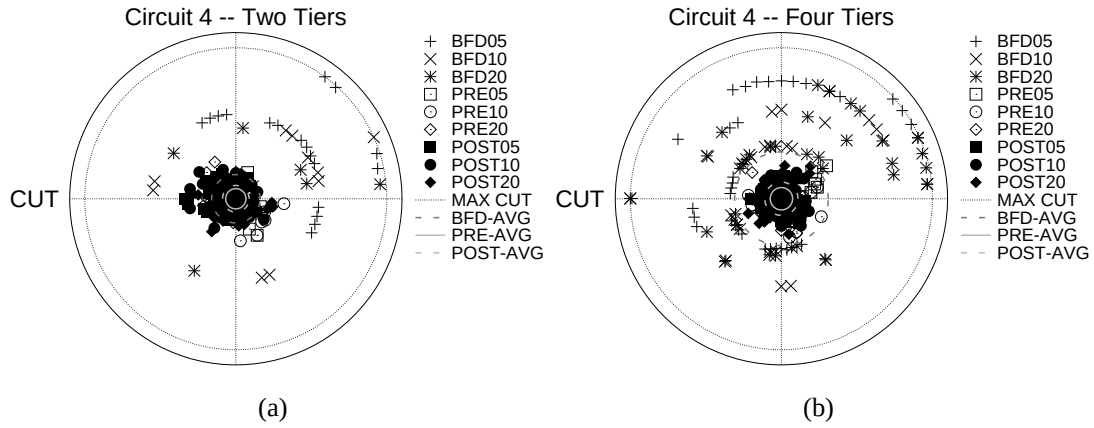


Figure 31. Polar plots of *cut* versus post-bond TAM width for *ckt4*. The four rings highlight the averages and max *cut*.

These tools are not perfect; at some design points the *cut* spikes up significantly. This is attributed to the second-class nature of the *cut* objective. Because our tool is designed to minimize the maximum wrapper chain length first, the *cut* is sometimes sacrificed to create a shorter wrapper chain. These outliers in the *cut* clusters can be used to inform the TAM architecture design; if wirelength or routing congestion are concerns in a particular wrapper design, assigning an additional test bit or two could help reduce the problem.

The other important point to note is that while the PRE and POST design tools both produce consistently low-*cut* wrappers, the PRE tool in general is the better of the two (6.6% on average compared to 8.4% for POST) as evidenced by the slightly tighter clustering of

the PRE results about the origins. We attribute this to the different scopes each method gives BFD and KL. The POST algorithm applies BFD to the global post-bond wrapper design problem and then KL to the local pre-bond wrapper design problems. In doing so, POST necessarily limits the opportunity for KL to optimize the pre-bond wrapper. In the worst case, every single scan chain in the post-bond wrapper would be stitched to scan chains from other tiers. This would leave KL with no opportunities to reuse the post-bond connections in the pre-bond wrappers. Conversely, the PRE tool applies BFD to the local problem and KL to the global problem. Unlike in POST, KL applied to the post-bond design problem is unrestricted by the physical layout of the stack and so is free to reuse any of the pre-bond stitches created by the BFD algorithm.

5.4 Summary

We have presented a methodology for designing 3D test wrappers for embedded 3D IP cores [42]. We use the *Best Fit Decreasing* and *Kernighan-Lin Partitioning* heuristics to design flexible test wrappers that can adjust to varying test modes like pre-bond and post-bond test. This flexibility results in a lower total test time for the CUT and reduced wiring resource consumption in the 3D wrapper design—the PRE design tool reuses 93% of the pre-bond stitching while sacrificing just 0.06% of the minimum possible test time. Our methodology is applicable to both true embedded 3D cores and to simpler planar embedded cores in cases where variable TAM bus widths are a useful design feature.

CHAPTER 6

SHORTING PROBE

The preceeding chapters have focused on testing the circuits internal to each unbonded tier. Testing these components is critical, since the majority of the design (tens of billions of devices and wires) resides within a tier. Testability was provided for these circuits right up to the 3D interface; our test architecture is able to verify test outputs sent to the 3D interface and source test inputs on the dangling input 3D vias. This functionality gets us most of the way towards complete fault coverage, but the 3D vias themselves, the metal blobs that actually form the microbumps and the TSVs, have so far escaped test. This is a problem because the 3D vias are subject to defects the same as any other component of the tier. A test methodology targeting the 3D vias specifically is required in order to completely test a 3D IC.

A variety of methods have been proposed for testing and characterizing 3D vias—Kelvin configurations and ring oscillators [75]; sense amplification [21, 80]; leakage monitors and capacitance bridges [46]. Unfortunately, all these techniques are designed for the post-bond test environment; they cannot function during pre-bond test because half of the test circuit is missing.

The sense amplification technique alone has been adopted for pre-bond test of 3D vias [16, 17]. Even then however, all these techniques are analog in nature, which is a problem. Analog test circuits are notoriously delicate, requiring finely tuned reference voltages and passive components and a very quiet operational environment (*i.e.*, little noise). Finely-tuned parameters are not at all cost-effective in a high-volume production environment, and digital ICs at very noisy chips. The techniques listed above also all rely on comparators, which are relatively large components that will not scale to the millions of sub-micron 3D vias we expect to see in near-future 3D designs.

Thus we require an all-digital test method that can be applied pre-bond to millions of

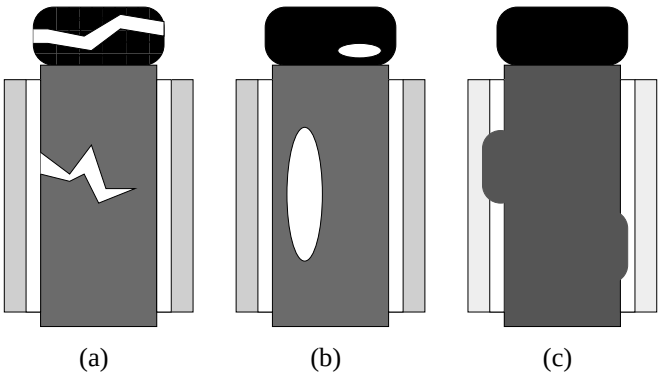


Figure 32. The three types of pre-bond-testable 3D via defects.

3D vias in a high-volume manufacturing environment. In this chapter, we present a new test methodology, Shorting Probes. This methodology utilizes the well-established technology of passive probes to test 3D vias with a high-speed scan-based methodology that can be easily integrated with current industry best practices.

6.1 3D Via Defects

3D vias are, just like any other feature of an IC, subject to manufacturing defects. Pre-bond, there are three different types of defects that may afflict a 3D via; these are illustrated in Figure 32.[1]

Figure 32(a) shows a break, a disconnect in the structure of the 3D via, caused some stress factor on the via. A break can occur in either the TSV, microbump, or the interface between the two. Similar to a break is a void, shown in Figure 32(b). A void is caused either by an incomplete fill of the TSV or the presence of a foreign particle in either the TSV or the microbump. Both defects increase the *through-resistance* of the 3D via.

In contrast, a pinhole, shown in Figure 32(c), is a resistive short to the grounded substrate. Pinholes are caused by a failure in the deposition of the insulating sheath that surrounds the TSV. These defects decrease the *ground-resistance* of the 3D via and make the attached node difficult to charge to a high voltage. Figure 32(c) shows two pinhole defects:

[1]The TSVs and microbumps in Figure 32 are shown at the same size for illustrative purposes only. The actual relative size of these structures is process-dependent.

a large defect on the left side of the TSV and a small defect on the right side. The width and the depth of the pinhole determine the severity of the defect.

The magnitude of these defects determines the resulting fault. A severe defect will create a stuck-open or stuck-at-zero fault in the 3D via node. A smaller defect will create a delay fault, impacting the speed of the 3D circuit. Detecting the stuck-at faults is the primary goal, while detecting the delay faults is important for establishing the timing margin on the 3D circuits (the smaller the detectable delay fault, the tighter the margin can be).

6.2 3D Via Probing

The fundamental challenge in trying to test 3D vias with standard probes is scale. Cutting edge 3D vias are currently being manufactured on a pitch of just a few microns, and sub-micron via technology is expected in the next couple years [64]. In stark contrast, the pitch of current test probe technology is about 100μm, and even advanced MEMS-based probe card technologies are only expected to push the pitch down to 40μm or so in the near future [73, 74]. This size discrepancy means that the 3D vias cannot be probed individually for the sake of test, so a traditional methodology cannot be used to test the vias pre-bond.

Rather than attempt to work around this size gap, we choose to use it to our advantage. We propose using traditional, large test probes to touch multiple 3D vias at once. By touching several vias simultaneously with a single test probe, we connect the vias together electrically (hereafter, we refer to the several 3D vias that share a single test probe as a *3DV set*), forming new circuit paths within the tier that can be used to test the vias for faults. Figure 33 provides an example. As shown, the tier under test contains two unrelated circuits. The circuit on the left is driving a signal to a neighboring tier, while the circuit on the right is receiving a signal, also from a neighboring tier. Pre-bond, both 3D vias are single-ended, lacking an observer and a controller, respectively. By touching these two vias with a test probe as shown in Figure 33(b), a new circuit path is formed, and so faults in the vias can now be controlled by the left circuit and observed by the right circuit, establishing

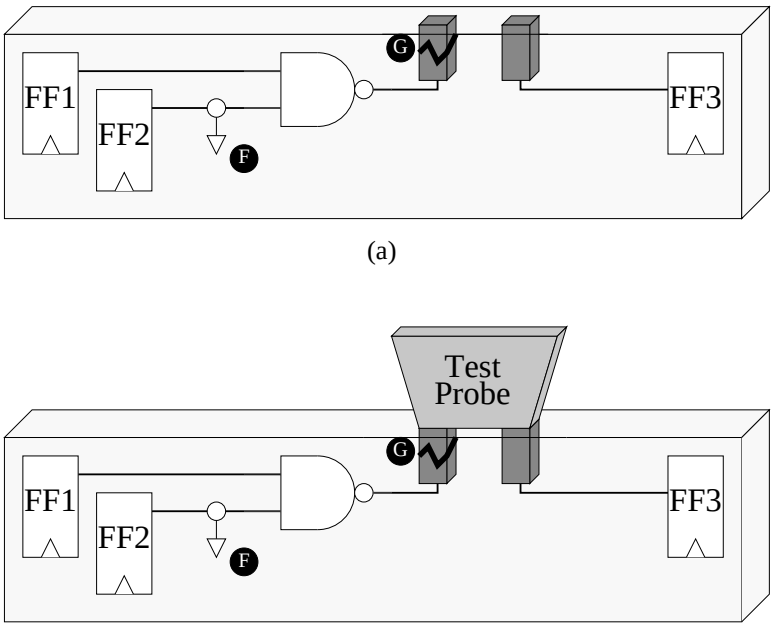


Figure 33. A pre-bond test scenario with faulty 3D circuits. Fault *F* is in-tier while fault *G* is in a 3D via. Both faults become testable when after the probe tip is used to create a new circuit path as shown in (b).

testability of these 3D vias.

Figure 34 shows a scanning electron microscrope (SEM) image of a next-generation test probe tip array. This particular array was jointly designed by Cascade Microtech and IMEC [74]. Its purpose is to contact a JEDEC 3D DRAM interconnect 3D via array [5] for pre-bond test. The tips are $6\mu m^2$ on a pitch of $40\mu m$. These tips are so small because a design goal of this array was to utilize the standard scrub-mark technique (contact the test point, then slide the probe tip laterally a short distance to decrease contact resistance) to contact the 3D via array, but different sizes and pitches are easily produced, according to the authors.

In our proposed technique, a similar MEMS probe array could be used, but with a pitch-to-width ratio much closer to two. The three key benefits of this style of probe tip array are the small size, low contact force, and tip planarity, all features not found in traditional probes. These features should enable the probing of 3DV sets as we propose.

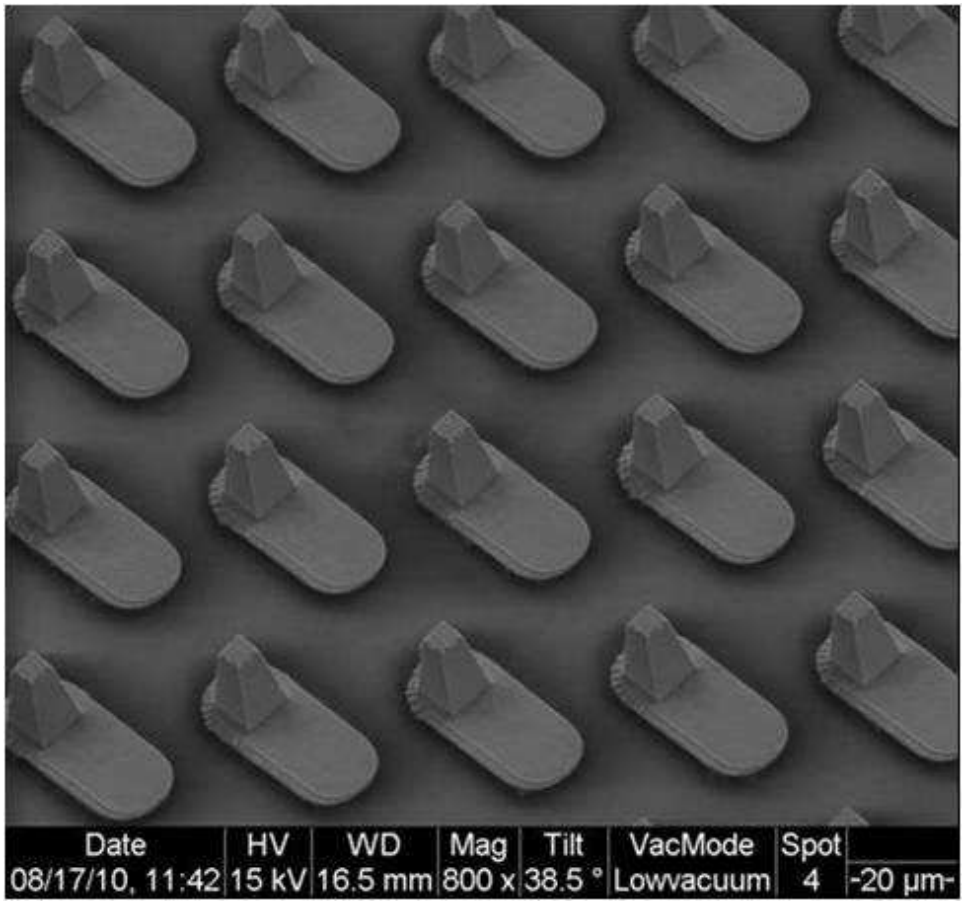


Figure 34. SEM image of a next-generation probe tip array fabricated with MEMS technology. *This image has been reproduced with permission from Smith* **et al.***, ITC 2011 [74]***.**

Noia and Chakrabarty [56] took a related but different approach to pre-bond 3D via test. They also proposed probing 3DV sets with traditional test probes. Unlike our methodology however, they assumed *active* circuitry—a reference capacitor and control logic—would be placed on the probe cards to measure the resistance and capacitance of the 3DV sets. There are two key problems with this approach. First, it is difficult to identify which 3D via in the set is faulty. Second, placing active circuitry on a probe card is non-standard, significantly increasing the complexity and cost of the cards.

We forego the active probe card circuitry and assume instead industry-standard passive probe cards. Their only purpose is to short neighboring 3D vias together and create new test paths. The scan chains are used to apply test vectors and recover test responses. This

general approach integrates seamlessly with the test procedures already in use in modern fabs.[2]

6.2.1 DFT Requirements

Implementing our proposed methodology puts some constraints on the physical design. The difficulty is that different 3D vias serve different purposes. Generally, each 3D via has one of the following purposes: rails, hardcore, signal drivers, and signal receivers. This diversity must be considered when the 3D interface is designed. Specifically, four DFT rules must constrain the 3D interface specification:

1. One driver and one receiver is required in non-rail, non-hardcore sets
2. If there is more than one driver in a set, tri-state functionality must be added to all drivers in that set
3. Rail 3D vias must be isolated within their own sets
4. Each hardcore signal must have a dedicated set

We will discuss each in turn.

Driver and receiver 3D vias carry the actual functional inter-tier signals within the 3D circuits and so are the primary test targets. Under our proposal, a single test requires applying a test vector from one driving circuit within each set to all receiving circuits within the same set by way of the test probe tip; this necessitates DFT rule #1. If zero drivers exist within a given set, a test-only driver must be added to provide the test signal source. Similarly, if there are no receivers within a set, a test-specific receiver must be added to observe the test response.

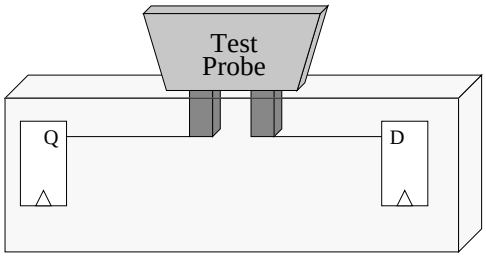
If multiple drivers exist within the set, all-but-one driver must be disabled during a given test to prevent contention, as specified in DFT rule #2. This can be achieved with the

[2]The actual act of probing 3D vias is different from traditional probing, as will be discussed in Section 6.5, but the methodology is the same.

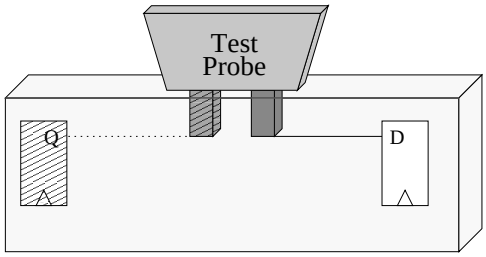
simple addition of a transmission gate to each driver: pre-bond this gate prevents contention between drivers; post-bond this gate is always enabled, completing the 3D circuit. If the driver in question already carries a tri-state signal, a simple hook in the enable logic is sufficient for our test methodology. These tri-state drivers can be coordinated with flip-flops that specify which driver is active in the set. Assuming a 100μm probe tip and 2μm-pitch 3D via, a maximum of fifteen flip-flops are required per set. That is a worst-case count of 15k flip-flops in a 10mm^2, a negligible overhead against the millions of flip-flops in a typical design. In the general case where sets are sparsely populated, this overhead drops to a few hundred flip-flops.

Rail 3D vias carry the VDD, ground, and other power rails across tiers. 3DV sets that include rail vias must be made up of only a single rail type (*e.g.*,all ground vias). Grouping a ground via with a VDD via would cause a high-current short, disabling the tier, while grouping any power via with a signal via would render the signal via untestable. This necessitates DFT rule #3 above as so constrains the design of the 3D power-delivery network. Fortuitously though, having sets of dedicated rail vias provides a ready method for powering up the tiers for pre-bond test.

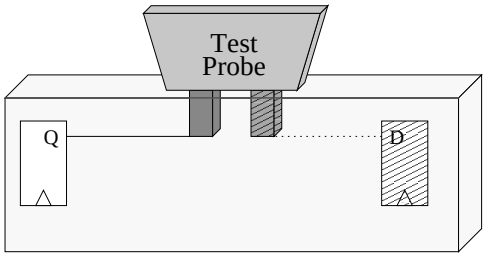
The hardcore consists of the control signals (e.g. clock, reset, and scan enable) required to manage test. These 3D vias are not themselves under test but rather carry the signals required to test the other vias. As such, similar to the rail vias, an entire 3DV set must be dedicated to each signal (e.g. one set for sourcing the clock and another to source reset), as specified in DFT rule #4. Generally these 3D vias will be left unused post-bond as the hardcore signals will have optimized 3D distribution networks [86]. The hardcore only consist of a few tens of signals at most. Assuming the same pitches as before, 25k hardcore vias are required for ten signals. By comparison, a small 3D stack with a 10mm^2 footprint can contain 2.5M 3D vias, so the overhead of these dedicated hardcore sets is negligible.



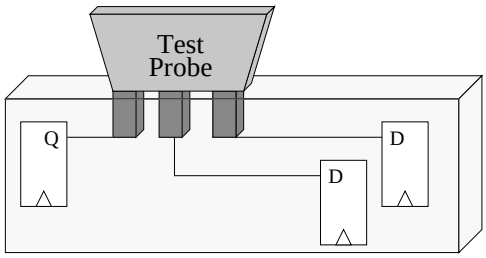
(a) Basic 3DV set: one driver and one receiver; no DFT required



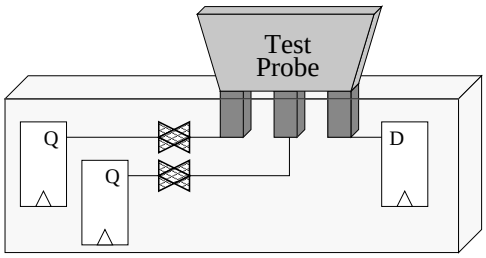
(b) No driver; DFT driver added



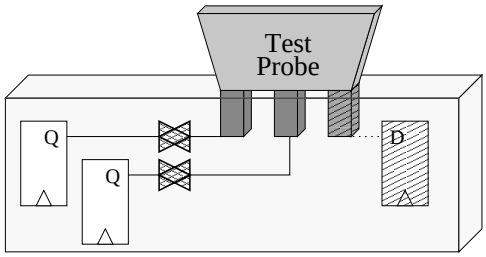
(c) No receiver; DFT receiver added



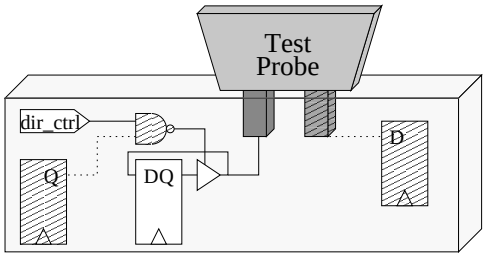
(d) One driver, two receivers; no DFT required



(e) Two drivers, one receiver; DFT pass gates added



(f) Two drivers, no receiver; DFT pass gates and receiver added



(g) Bidirectional source; DFT receiver and tri-state control added

Figure 35. Shown are a variety of possible 3DV sets. The number of driving and receiving 3D vias in a given set determine the required DFT structures. The additional DFT structures that must be added in each example are shown hashed.

6.2.2 DFT Example

Figure 35 highlights several possible types of 3DV sets. Figure 35(a) is the basic set with one 3D via driving an output and the other via receiving an input. Per rules #1 and #2, no DFT is required for this set. Figures 35(b) and 35(c) show the basic set but missing the driver and the receiver respectively. Rule #1 requires that these functionalities be replaced, as shown by the hashed structures. Figure 35(d) shows a one-driver-two-receiver 3DV set. Just like the basic set, no additional DFT is required. Figure 35(e), in constrast, has two drivers, so as required by rule #2 pass gates have been added to prevent conflicts during pre-bond test. Figure 35(f) has two drivers and no receiver, requiring the addition of both an observing flip-flop and pass gates; this type of 3DV set is the worst case in terms of DFT overhead.

Figure 35(g) shows a special case application of rules #1 and #2, a bi-directional 3D via. Note that in the application of rule #1, a bi-directional via may serve as either the driver or the receiver but not both (*i.e.*, self-test is not allowed) because using it in both capacities would result in testing only the net attached to the 3D via, not the via itself. In Figure 35(g) then, the bi-directional 3D via serves as a driver and so a DFT observer is added. For rule #2, the bi-directional circuit already has tri-stating capability which can be used to prevent conflict with another driving 3D via (not shown in the figure). Therefore, to satisfy rule #2, we add a DFT hook into the enable signal logic (represented by the NAND gate) to allow the enable signal to be controlled by both the functional path (*dir_ctrl* in the figure) and the test path.

In general, 3DV sets will be composed of square arrays of 3D vias, not lineary arrays as has been shown. Figure 36 shows a more complex illustration of our proposal using a square 3DV set. In the figure, four 3D vias have been shorted into a set. The two left vias are drivers and the two right vias are receivers; thus this set has already satisfied DFT rule #1. Because this set has more than one driver, DFT rule #2 requires that the drivers have tri-state functionality. To satisfy this rule, pass gates have been added to the driving

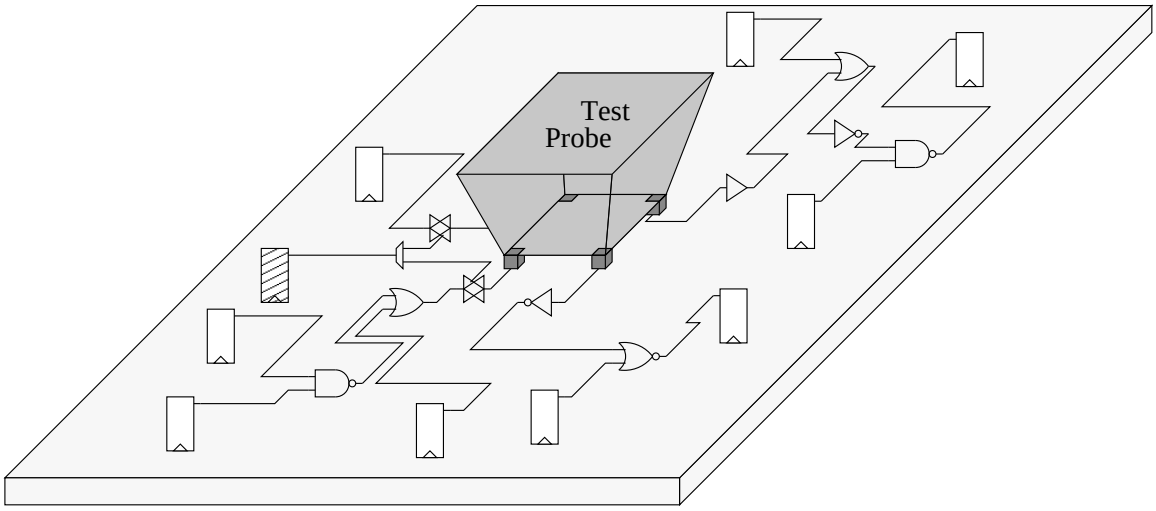


Figure 36. An example of the application of our test methodology to an unbonded silicon tier. Four circuits have been connected by the test probe.

circuits, and a counter and demultiplexer have been added for control (in this example, it is a one-bit counter represented by the hatch-filled flip-flop). All the shown flips-flops would be included in the scan chain; this connection is not shown for figure simplicity.

An example 3D interface is shown in Figure 37(a) to illustrate DFT rule #3 and #4. The rail 3DV sets (for VDD and GND, in this example) are placed regularly across the tier. The driver and receiver 3D vias are placed in between the power stripes. The hardcore 3DV sets are placed off to the side to minimize their impact on the performance of the 3D circuits, just as test control circuits (e.g. IEEE 1149.1 taps) are placed in non-critical locations in traditional planar design. Such a design supports the various power delivery networks, provides the necessary test control, and minimizes the constraints on the placement of the signal 3D vias.

6.2.3 Test Insertions

Test probes have a minimum width and pitch, and generally the pitch must be at least twice the width. This pitch constraint means that two adjacent 3DV sets cannot be probed in the same test insertion; at least two are required. This constraint has little impact on the drivers and receivers; they are simply probed in the insertion in which they are reachable. The

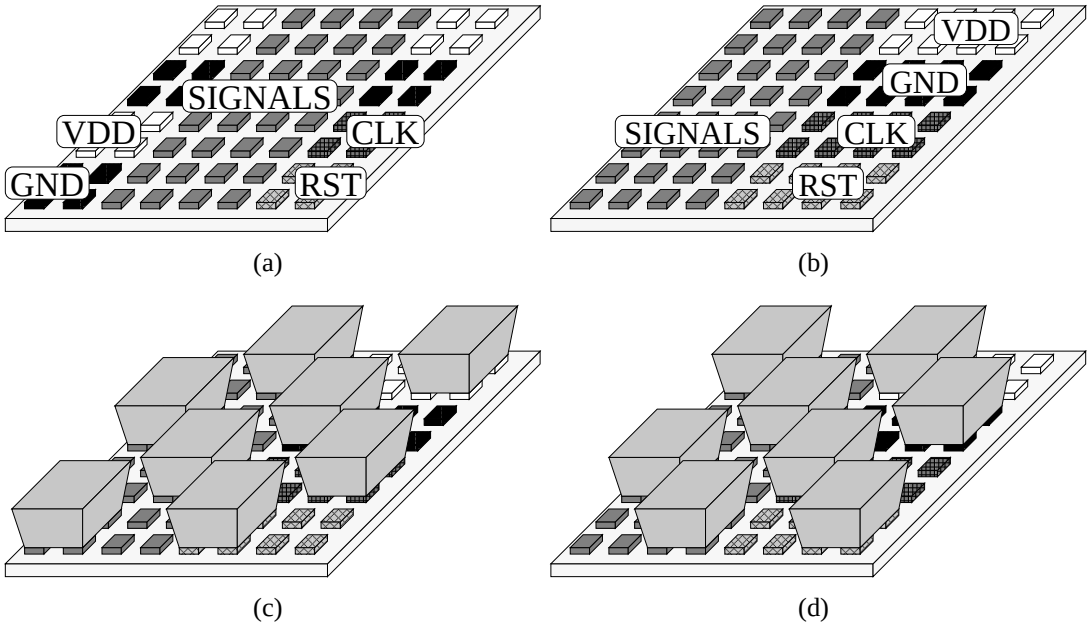


Figure 37. A generalized 3D via assignment plan.

placement of the rail and hardcore 3D vias however must account for the several insertions. To minimize the test cost, it is preferred to have only one probe card design, requiring the reuse of that design across all insertions. That requires a tweak to the 3D interface design, as shown in Figure 37(b). In this design, the rail and hardcore sets have been doubled up. One set is used for the first insertion (Figure 37(c)) and the second set for the second insertion (Figure 37(d)). This allows a single probe tip arrangement to power up the tier and driver the hardcore in both insertions.

Necessarily, only a fraction (half, in the example of Figure 37) of the rail sets are driven in a given insertion. This limits the power draw allowed for pre-bond test to what can be supplied by these sets. Generally, rail 3D vias should be over-provisioned to minimize IR-drop and $\frac{di}{dt}$ problems within the 3D stack. If not, standard test-power-reduction techniques can be employed to reduce current draw. Note that only half the 3DV sets are under test in a given insertion. Therefore an easy power-reduction technique would be to not activate the sets not under test.

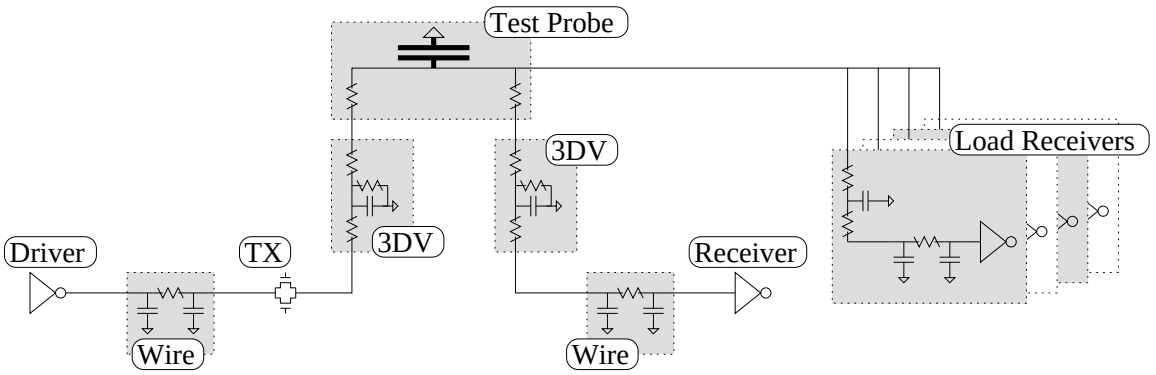


Figure 38. The circuit model of the 3DV set test system. The model components, from left to right, are the driver and its wire, the transmission gate, the driving 3D via, the probe tip, the receiving 3D via, the receiver and its wire, and a set of loading circuits representing other vias in the set.

6.3 Experimental Setup

6.3.1 Modeling

To evaluate our proposed DFT scheme, we simulate the test circuits created by the probe tips. Our circuit model is shown in Figure 38. The model is composed of four main components; from left to right in Figure 38, these components are the driving circuit, the test probe, the receiving circuit, and the load circuits. The driver is the source of the test signal, and the receiver is the observer of the test signal. The load circuits model the additional circuits in the 3DV set, and the test probe completes the test path. Additional drivers within the set are not modeled because the output capacitance of their transmission gates is negligible.

The driving circuit is composed of a driving buffer, a wire, a transmission gate, and a 3D via. The buffer and the transmission gate are simulated using the high-performance 32nm transistor models from the Predictive Technology Model [7]. For the wire we use a π-model, taking the resistance and capacitance values from the PTM as well. For the 3D via we use the model developed by Katti et al. [35].

The receiving and load circuits have the same basic form as the driving circuit, minus the transmission gate because they cannot contend with the driver. The test probe is represented with a T-model. The two resistors model the contact resistance between the probe

Table 9. The list of circuit model parameters and the associated default value.

Parameter	Default
Drive Buffer Size	16x
Drive Wire Resistance	60Ω
Drive Wire Capacitance	7fF
Transmission Gate Size	16x
Drive via Resistance	0.1mΩ
Drive via Resistive Ground	1MΩ
Drive via Capacitance	16.6fF
Driver Contact Resistance	0.1Ω
Probe Capacitance	2pF
Receiver Contact Resistance	0.1Ω
Receive via Resistance	0.1mΩ
Receive via Resistive Ground	1MΩ
Receive via Capacitance	16.6fF
Receive Wire Resistance	60Ω
Receive Wire Capacitance	7fF
Receive Buffer Size	16x
Number of Load Circuits	2
Load via Resistance	0.1mΩ
Load via Capacitance	16.6fF
Load Wire Resistance	60Ω
Load Wire Capacitance	7fF
Load Buffer Size	16x

and the driving and receiving 3D vias, respectively. The capacitor represents the load of the probe tip itself, which must be charged by the driver. We do not model the resistive and inductive characteristics of the tip because our test methodology only requires the probe tip, not the entire cable assembly that normally connects the probe to the test equipment. The two resistors tied to the probe tip represent the contact resistance between the probe tip and the 3D vias. This will be examined in detail in Section 6.5.

6.3.2 Parameters

The circuit parameters and the associated default values in our model are listed in Table 9. The default wire resistance and capacitance values are taken from a 3μm wire. The default 3D via resistance and capacitance values are extrapolated from the 3D via modeling work in [35] and [37]. The number of load circuits is based on the 3D connection density from

Table 10. The list of variables in the sensitivity analyses.

Component	Range
Drive Buffer Size	2x – 80x
Drive Wire Length	$0.1\mu m$ – $1000\mu m$
Receiver Buffer Size	2x – 80x
Receiver Wire Length	$0.1\mu m$ – $1000\mu m$
Load Buffer Size	2x – 80x
Number of Load Circuits	1 – 32

the 3D multiprocessor system presented in [27]. The probe capacitance is based on the products offered by Cascade Microtech [15]. The contact resistance is taken from [74].

6.4 Results

Here we report the results of our simulations. We conduct two different experiments in our evaluation. First, we conduct a series of sensitivity analyses to investigate the impact of different parameters on testability. Second, we use a Monte Carlo simulation to examine the effect of varying all the parameters together.

6.4.1 Sensitivity Analysis

For our sensitivity analyses, we vary the strength of six different circuit parameters, one at a time. These are listed is Table 10. Note that buffer size ranges listed are multiples of the minimum width. We examine buffers of different sizes in the driver, receiver, and loads because there is no guarantee that strengths of the circuits grouped into a set will be well matched the way they are in a normal circuit design. We also vary the length of the wires connecting the 3D via to either the driver or the receiver. We vary this parameter independently of the buffer strength because we have no control over the partitioning of the 3D circuit and so cannot guarantee that these two parameters are matched. For example, if most of the 3D circuit is on the neighboring tier, the 3D-via-under-test may have a large driver attached to a short wire. Conversely, if the neighboring tier contains just the receiving flip-flop, the 3D-via-under-test might be driven by a relatively weak driver and long wire

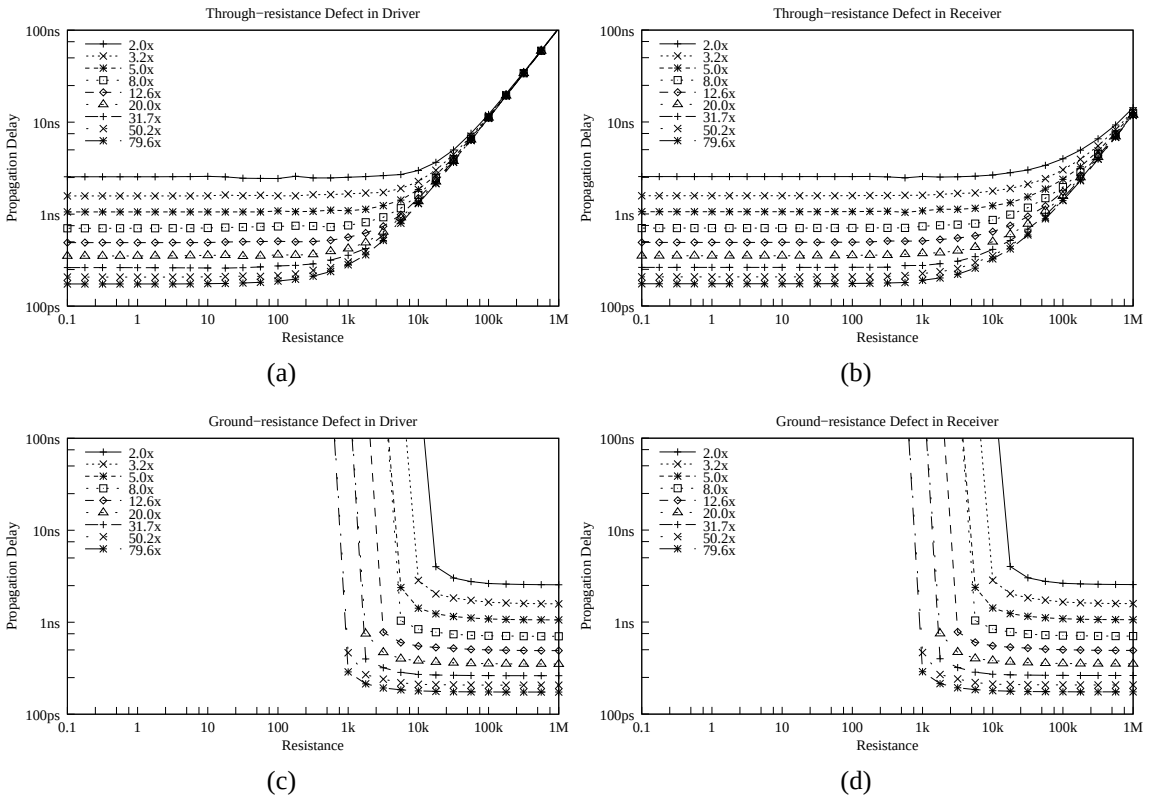


Figure 39. Propagation time results for varied driver widths.

because of the minimal after-3D-via load. Finally, we vary the number of load circuits in the set to test the sensitivity of our methodology to the density of 3D interconnects.

6.4.1.1 Sensitivity to Driver Width

Figure 39 shows the propagation time of a signal through a 3D via set plotted against the resistance of the fault. The several curves represent the increasing widths of the driver. Propagation times for through-resistance defects in the driving 3D via are shown in Figure 39(a), for ground-resistance defects in the driving 3D via in Figure 39(c), for through-resistance defects in the receiving 3D via in Figure 39(b), and for ground-resistance defects in the receiving 3D via in Figure 39(d). Note that these results are log-log plots.

First consider the results for the through-resistance defects. There are two regimes apparent in the graphs. On the left is a near-constant response; this means that the driver is strong enough to overcome the relatively low-resistance defects (1Ω–10kΩ). Then there is

a distinctive knee point where the response becomes linear. Now the defect is dominating the circuit, and the resulting stuck-open fault is easily detectable in the receiving circuit. Effectively, this knee point defines the smallest detectable defect. Note that as the driver size increases, the knee resistance decreases and so smaller defects can be detected. This is a DFT opportunity in that large drivers could potentially be incorporated into sets to increase defect detection.

The other pair of graphs in Figure 39 reports the propagation time when the circuit is beset by a ground-resistance defect. The near-vertical lines indicate the resistance at which the circuit was first able to drive the receiver high[3] within the 500ns simulation period. So, for example, with a 2x driver and a ground-resistance defect in the driving 3D via, the grounding resistance must be at least 32kΩ to propagate the high voltage successfully. This means that stuck-at-zero faults are easily detected pre-bond. As with the through-resistance defects, detecting small-leakage faults in the 3D vias is a matter of driver size.

6.4.1.2 Sensitivity to Other Variables

The sensitivity results for the other variables listed in Table 10 are shown in Figures 40 through 44. Figure 40 shows the circuit's sensitivity to the length of the driving wire. Notably, the circuit is sensitive to the wire length only up to a point ($\tilde{1}00\mu$m). All shorter wires show effectively the same response trend. This is an encouraging result because it means a large, test-specific driver that is inserted to test small-delay faults (as suggested by the driver strength results) does not need to be placed immediately adjacent to the 3D via; designers have the freedom to place it up to 100μm away without impacting fault detection capability. This freedom will significantly reduce the impact of this DFT method on the functional circuit performance.

The results for the receiver strength (Figure 41) and receiver wirelength (Figure 42) show that the circuit is almost completely insensitive to these variables. This is expected

[3]Ground-resistance defects are generalized stuck-at-zero faults, so the appropriate test pattern is to drive the net high

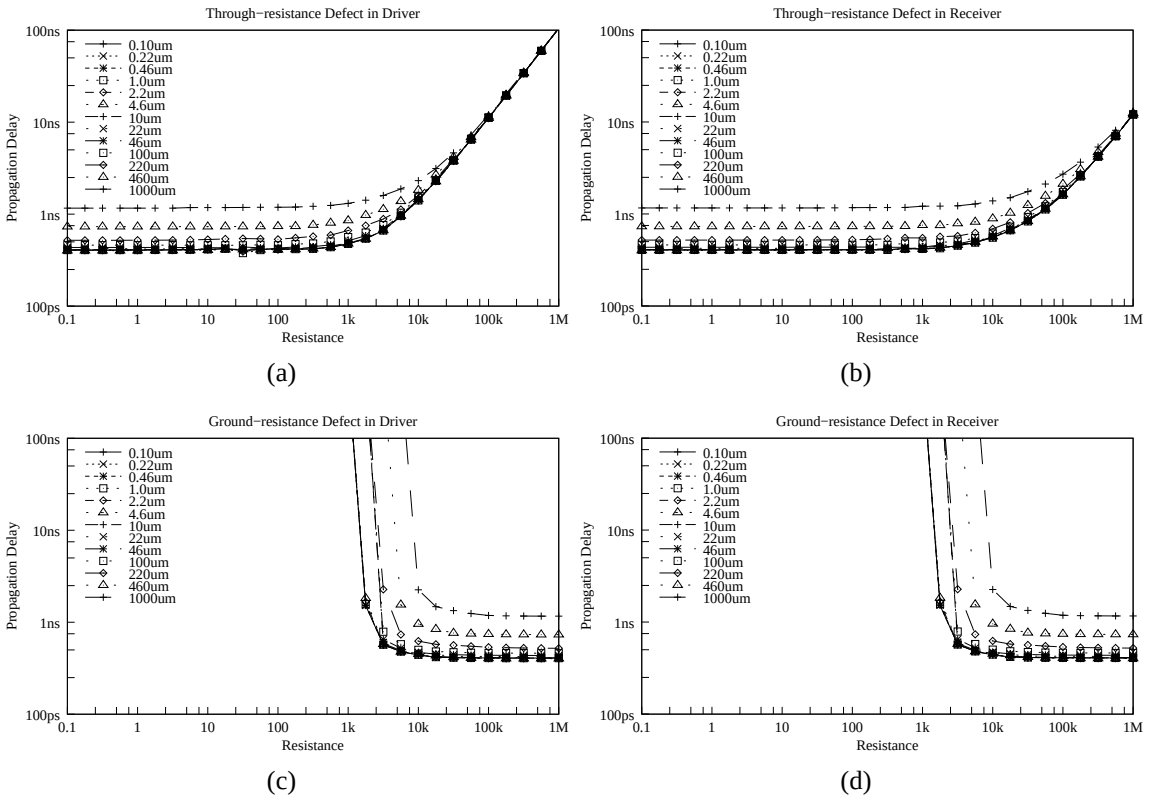


Figure 40. Propagation time results for varied lengths of the driving wire.

because these loads are negligible compared to the large probe-tip capacitance which precedes them. In all but the most extreme case (this case being a strongly-open fault combined with a very large receiver, as shown in Figure 41(b)), the receiving nodes charge just as fast as the 3D via node, unhindered by the small weights of the attached components.

The story is the same for the load size sensitivity (Figure 43); a large load-receiver is negligible compared to the weight of the probe tip. This is not the case for the number of loads (Figure 44[4]). Rather, the parasitic capacitance of the loading wires adds to the weight of the probe tip, increasing the propagation time. As we would expect, the larger the number of loads, the more severe the effect. With five or fewer loads (for a total of seven 3D vias in the set), the impact of the loads is negligible. This is good because is typical designs like [36], the number of loads does not need to be a design concern.

[4]Figure 44 reports results at fractions of a load because the loads are simulated as lumped-sum elements, not as individual circuits. This does not impact result accuracy.

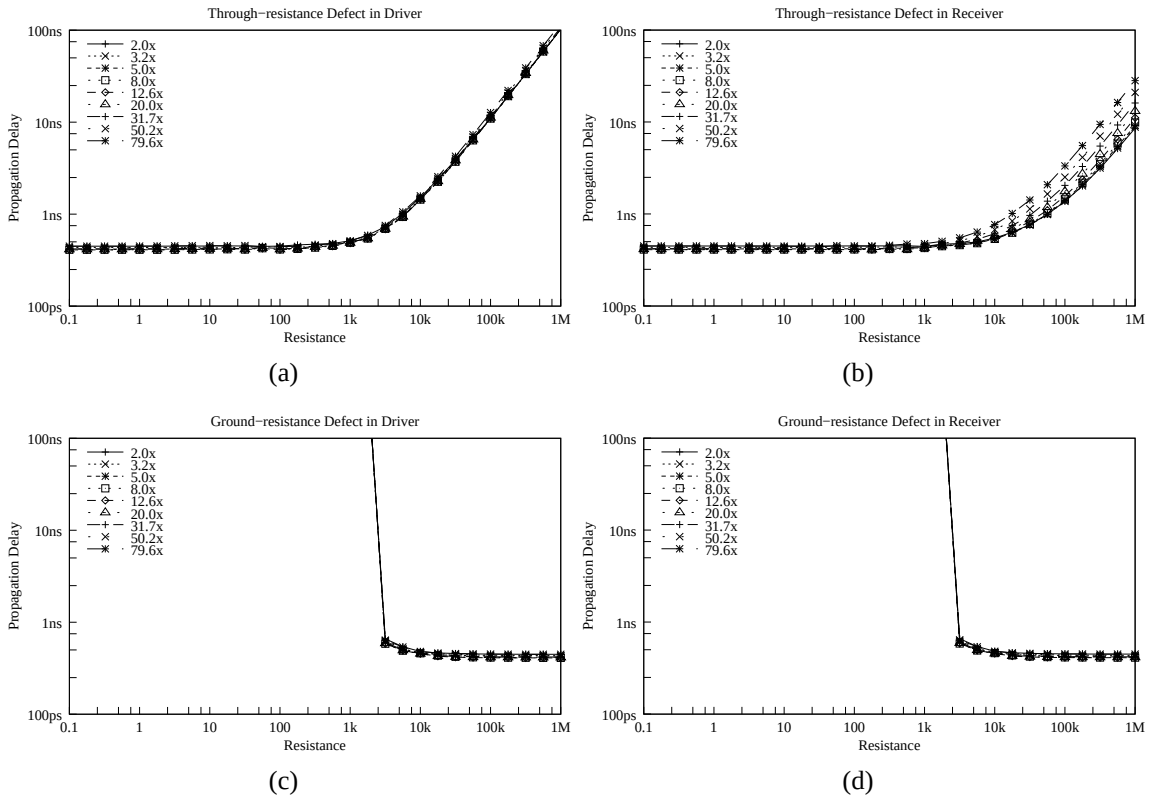


Figure 41. Propagation time results for varied receiver widths.

6.4.1.3 Compiled Sensitivity Results

The results for all experiments are summarized in Figure 45. For the through-resistance defect simulations (Figure 45(a) and (b)) the knee points in the data trends are reported. For example, a through-resistance fault in the driving 3D via creates a knee at 21kΩ for a 2x, but this point drops to just 1.6kΩ for an 80x driver. For the ground-resistance defects (Figure 45(c) and (d)) the turn-on points, the resistance at which the circuit was first able to successfully propagate the high signal, are reported. For example, with a 2x driver the grounding defect had to be at least 32kΩ for the circuit to operate, but with an 80x driver, even a 1kΩ defect could be overcome.

There are a couple important trends to note here. First, as we would expect, increasing the circuit strength (e.g. the driver and receiver sizes) increases the resiliency of the circuit

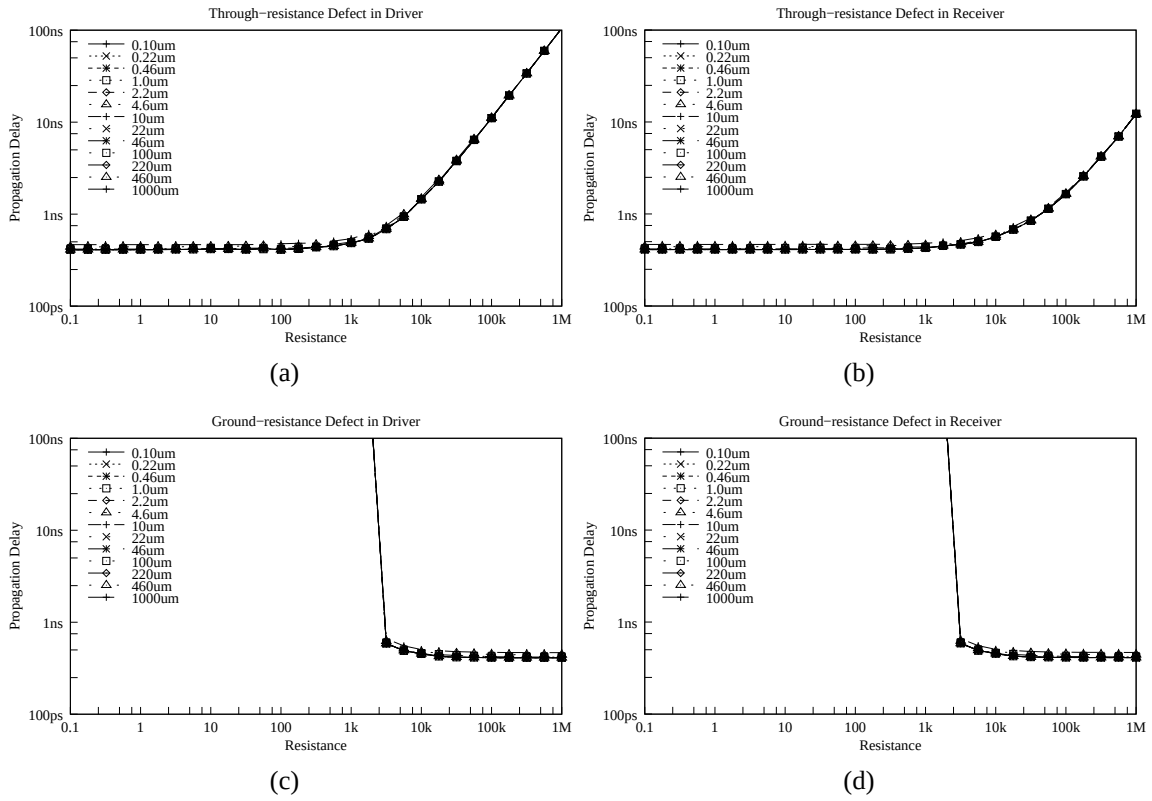


Figure 42. Propagation time results for varied lengths of the receiving wire.

to defects, while increasing the load factors (e.g. the wire length, load size[5], and number of load circuits) makes the circuit more susceptible to defects. Second, some components (driver size, driving wire length, and number of loads) are much more important factors in determining the circuit response than others (receiver wire length and the load buffer size).

The receiver size is a interesting component, as it has little effect on the circuit response to a through-resistance defect in the driving 3D via but significantly affects the response to a defect in the receiving via. This difference can be attributed to the ordering of the defect and the large probe tip capacitance. When the defect precedes the probe (in the case of a driving 3D via defect), the receiver can do nothing to help the driver charge the probe tip faster. However, when the defect follows the probe tip (in the case of a receiving 3D via defect), a larger receiving buffer is able to respond to the weak incoming signal strongly

[5]A large load buffer does decrease the propagation time of the test signal to the load output. However, we are interested in the effect of the larger buffer on the receiver output, which the larger load buffer sizes harm.

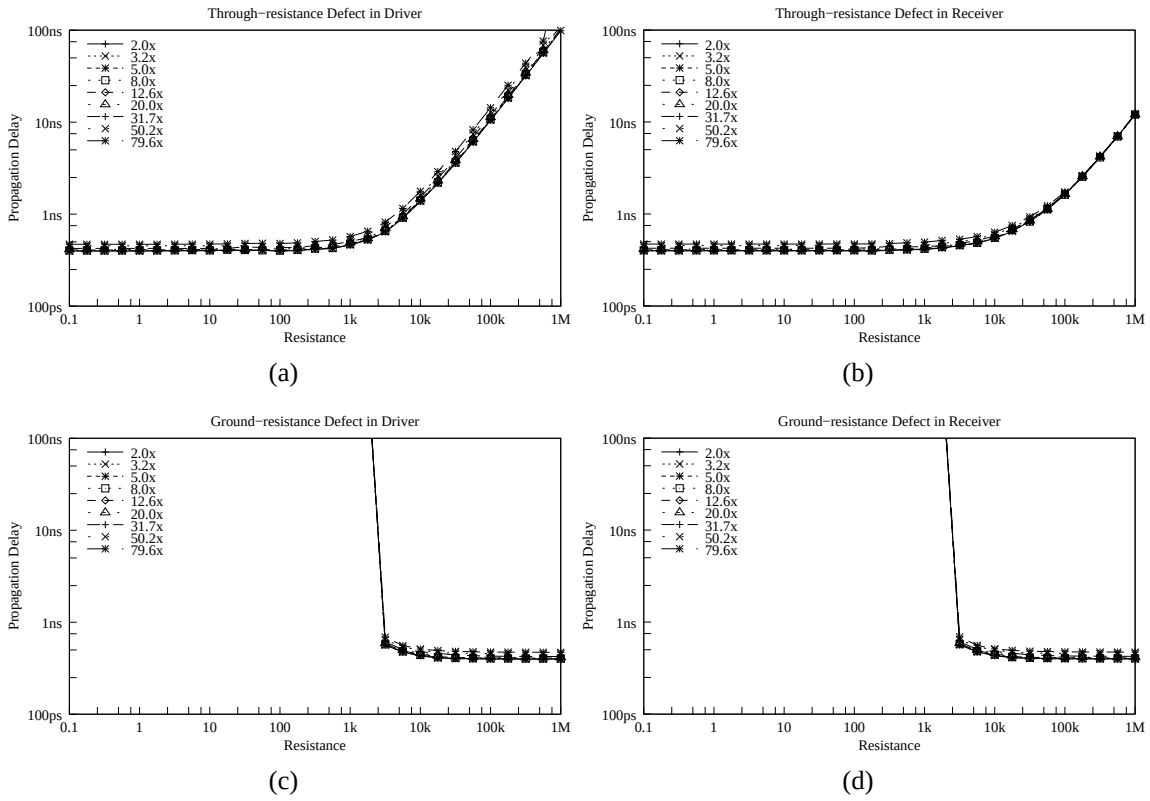


Figure 43. Propagation time results for varied load widths.

and so significantly reduce the propagation time. This is a significant result because the differing responses mean that driving and receiving 3D via faults are distinguishable. A fault in the driving 3D via will impact the test response of all receivers, while a fault in the receiving 3D via will impact only the response of that receiver. Depending on the resiliency and repairability of the circuits involved, the ability to distinguish between the two faults could be critical in correctly identifying the tier as good or bad.

For the ground-resistance faults, it is interesting to note that the circuit responses to both defects are identical; it does not matter whether the fault occurred in the driving 3D via or the receiving 3D via. This is because our model does not account for the resistance from one 3D via to the other through the probe tip—in practice the response to the two defects would differ slightly. However, the probe tip is very low resistance because it is a short, wide path, so its impact will be quite small, hence our decision not to model it. What

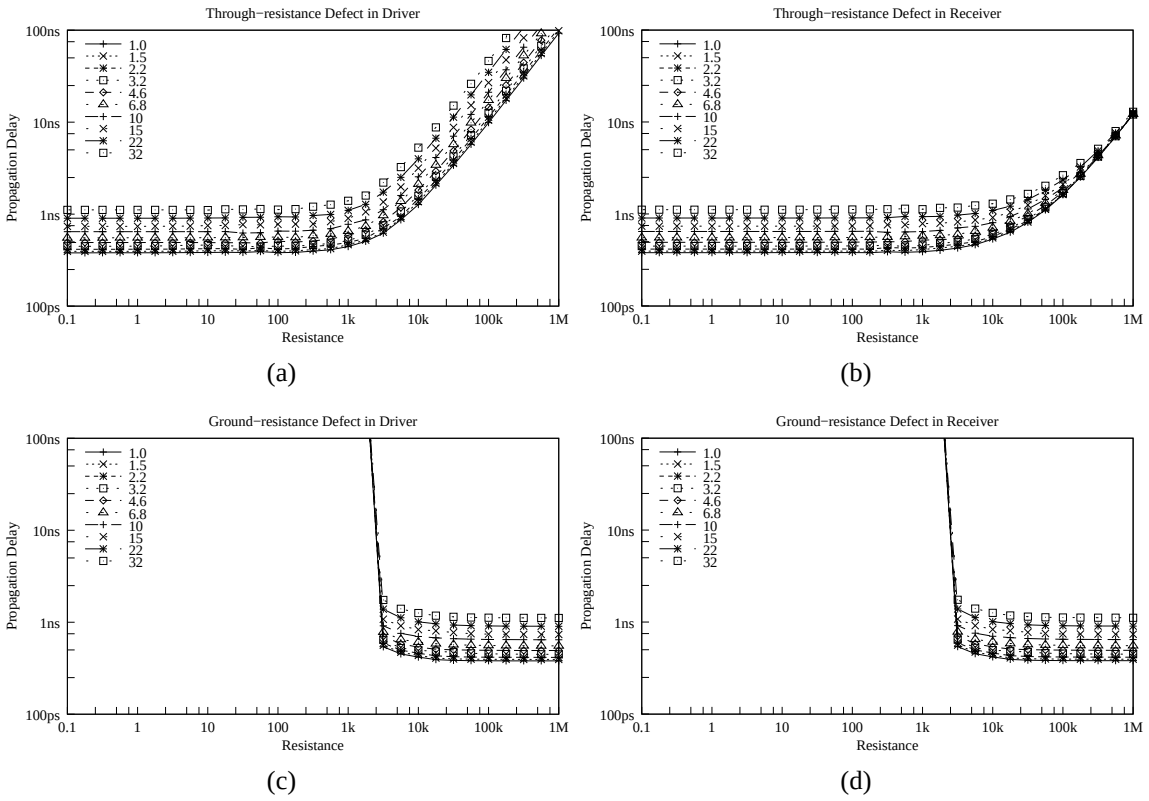


Figure 44. Propagation time results for varied numbers of load circuits.

this means in practice is that, unlike through-resistance defects, ground-resistance faults will likely not be distinguishable; our methodology, while able to detect the stuck-at-zero fault in this 3D via set, would be unable to determine whether the fault occurred in the driving or receiving 3D via. Unfortunately, switching to another driver would not help, as the resistive ground defect exists after the transmission gate that could otherwise be used to isolate it from the set. Note that we could distinguish these two faults using additional test insertions to separate the drivers and receivers into different sets. However, the resulting cost increase from greater test time, probe card degradation, and risk to the tier under test makes such an approach impractical.

6.4.1.4 Impact of Probe Technology

In the previous two sections, we analyzed the impact of circuit variables that chip designers control and which can be manipulated by the tool flow to increase fault detection. However,

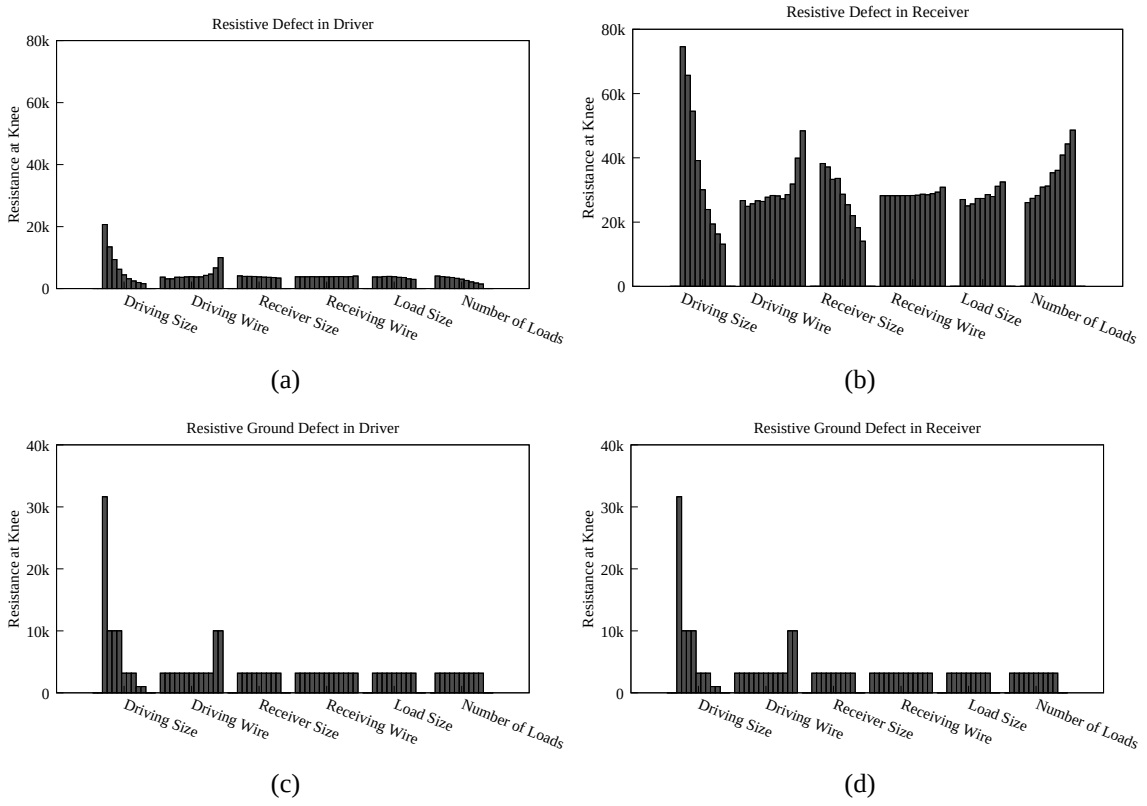


Figure 45. Plot of the knee points for all the simulated variables.

there is one critical circuit parameter which is well beyond the control of the design team: the capacitance of the probe tip.

Figure 46 reports the sensitivity of the 3DV set to the probe tip capacitance. We vary the capacitance from 10fF to 10pF to cover the spectrum of current and near-future probe technologies. For comparison, a mass-market probe tip has a capacitance of approximately 7pF, and a state-of-the-art probe tip has a capacitance of approximately 2pF. The MEMS-based probe tip discussed in Section 6.2 has a capacitance down around 100fF. Alternatively, Figure 47 reports the knee and turn-on points explicitly.

First, we note that the probe tip capacitance has a strong impact on the propagation delay, stronger than any circuit parameter examined in the preceding section. This means the probe tip technology is critical to test performance. Specifically, improving the probe tip technology can reduce the propagation time and therefore test time by a factor of nearly

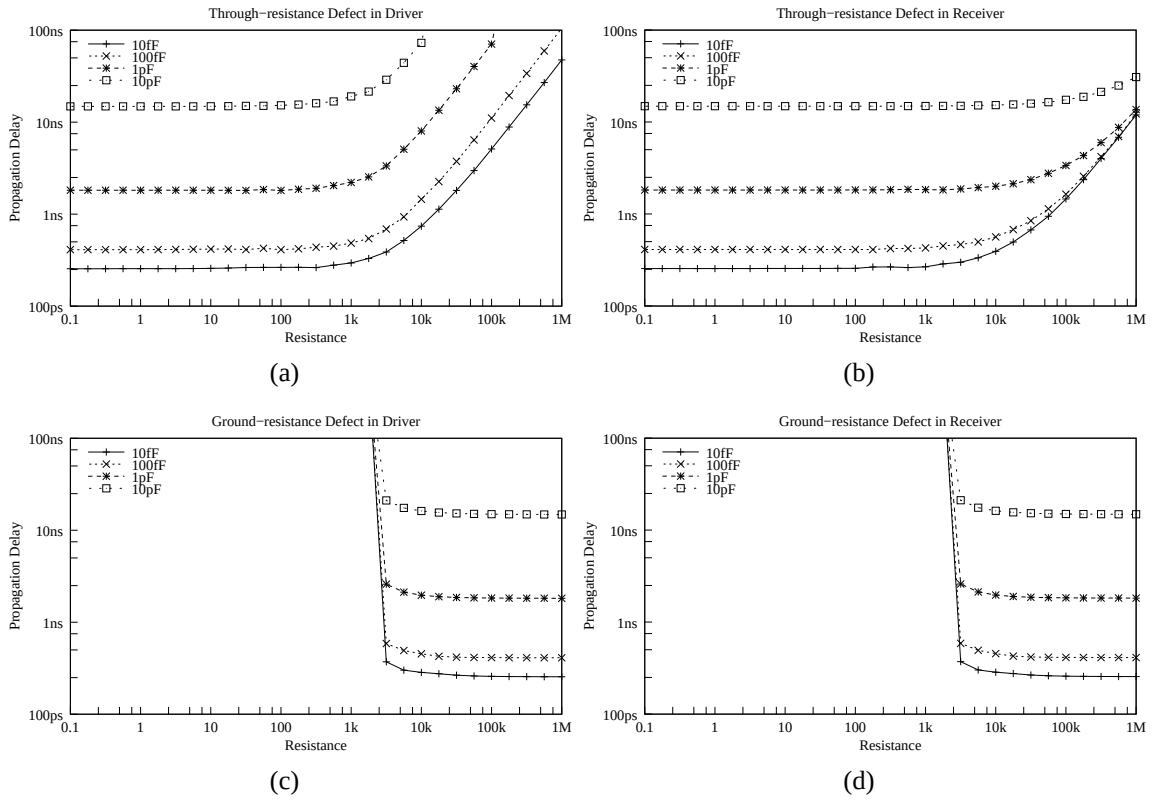


Figure 46. Propagation time results for varied capacitances of the probe tip.

10x (*e.g.*, a test machine equipped with 1pF probes can test ten paths in the time it would take a 10pF machine to test just one). This test time reduction would make a strong case for the deployment of advanced test probes in 3D fabs.

More interesting, however, is the relation between the probe tip capacitance and the knee and turn-on points in the resulting curves. For the ground-resistance faults (Figures 46(c) and 46(d)), the turn-on points do not deviate from the 3.2kΩ value seen for the other circuit parameters. This is because the turn-on point is defined by the resistance at which the ground-path is able to dissipate charge faster than the driver can source it, not on the size of the capacitor being charged. That is, the charging of the 3DV set is determined by the balance between the RC delay of the charging circuit and the RC delay of the grounding circuit; since the capacitance is the same in both, the probe tip capacitance does not affect the turn-on point.

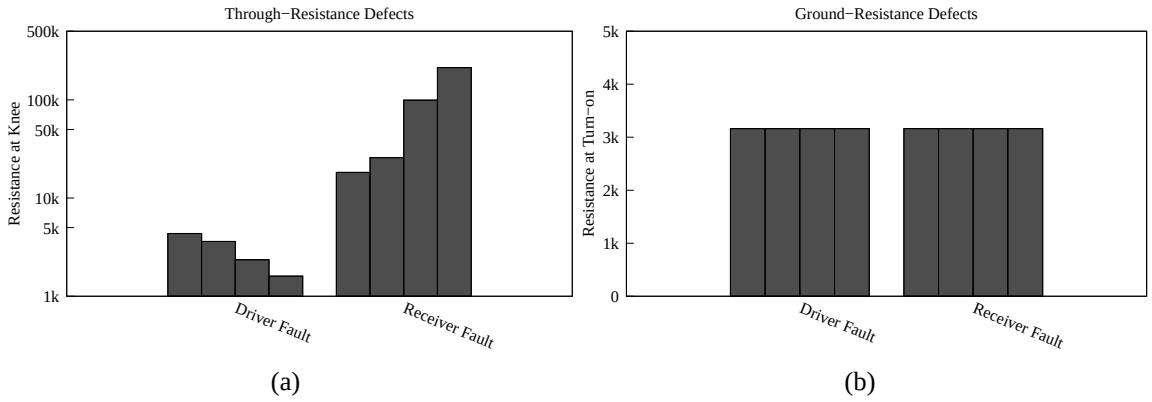


Figure 47. Knee and turn-on point results with increasing probe capacitance. Note that a log scale is employed for the y-axis in (a).

For the through-resistance faults, the location of the fault—driver or receiver—has a big impact. For a fault in the driver (Figure 46(a)), a larger probe capacitance increases the propagation time but does not significantly increase the knee point resistance. This is because the knee point is determined by the ratio between the resistance of the driving circuit and the through-resistance of the via. For small faults, the driving circuit dominates; for large faults, the through-resistance dominates. The magnitude of the probe tip capacitance has no bearing on this ratio, so it does not affect the knee point. In contrast, when the through-resistance fault is in the receiver (Figure 46(b)), the probe tip capacitance affects both the propagation time and the knee point. This is because a larger probe tip slows the charging of the 3DV set, while a larger through-resistance fault in the receiver slows the charging of the receiver node. This means that the response due to a large probe capacitance is indistinguishable from the response due to a large through-resistance fault. Therefore, smaller through-resistance faults are exposed when using a small probe but hidden when using a larger probe. So to increase coverage of smaller through-resistance faults in the receivers, smaller probes must be used.

6.4.2 Monte Carlo Simulation

We have evaluated the effect of each parameter on circuit performance, but the cumulative effect of these varying parameters is more important. To evaluate the impact of all the

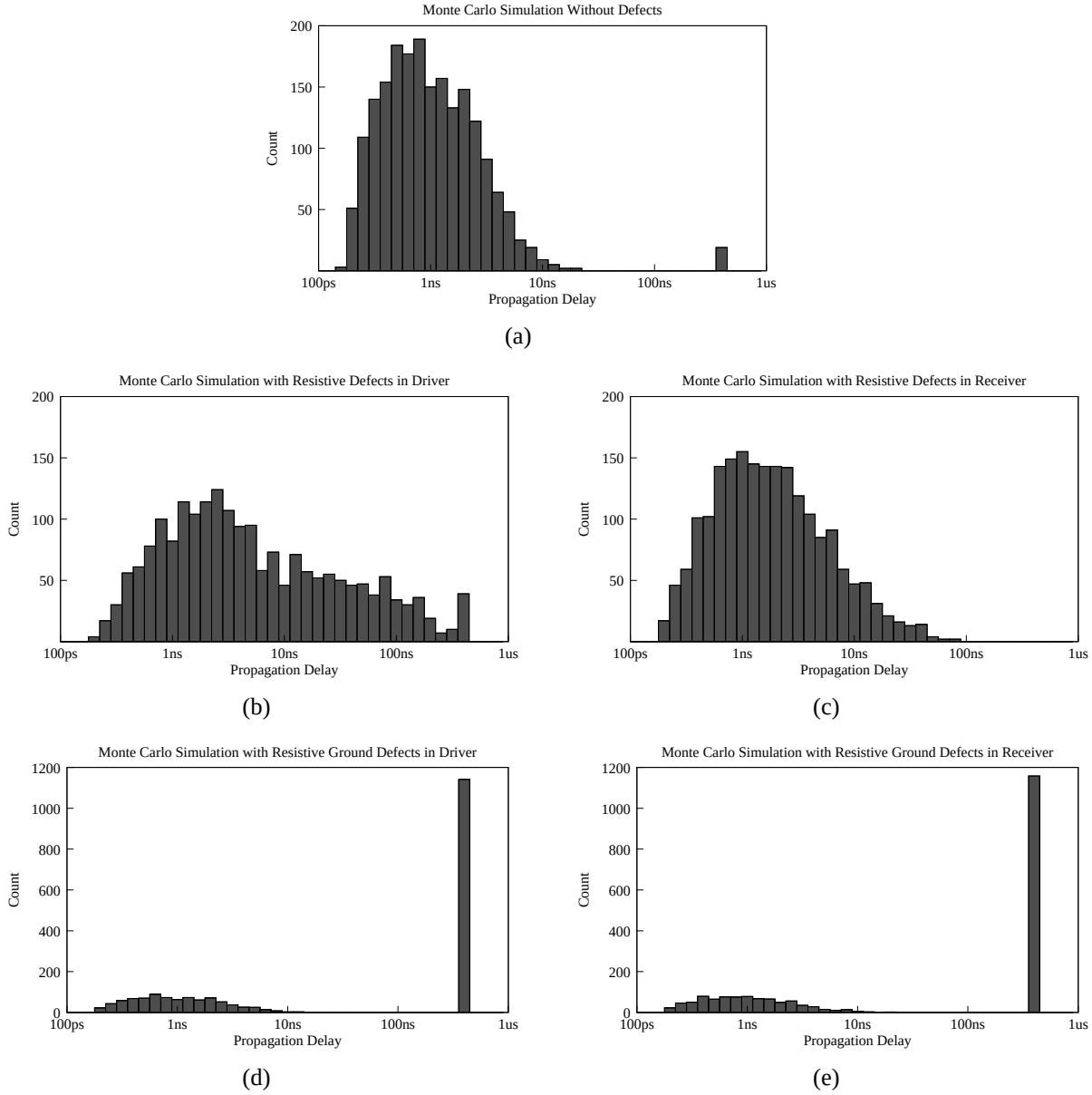


Figure 48. Distribution of the response times in the Monte Carlo simulations.

Table 11. Average and standard deviations of the Monte Carlo simulations.

Experiment	-σ (ns)	**Mean** (ns)	+σ (ns)
No defect	0.39	1.14	3.39
Driver, resistive	0.96	5.89	36.1
Driver, resistive ground	1.66	36.3	792
Receiver, resistive	0.63	2.01	6.38
Receiver, resistive ground	1.74	38.0	829

variables at once, we ran five Monte Carlo simulations, one each for each defect type and a fifth simulation for the defect-free case. Each simulation consists of 2000 data points for a total of 10,000 experiments. In addition to the six variables from the sensitivity analyses, we also allow the length of the load circuit wire to vary. We use the same parameter range as before, and we use an exponentially uniform distribution (e.g. the probability of choosing a wire length in the range $1\mu m$ to $10\mu m$ is identical to the probability in the range $10\mu m$ to $100\mu m$) to pick the sample points. The defect resistances are also selected from an exponentially uniform distribution.

Figure 48 presents the results. Figure 48(a) shows the defect-free propagation times, Figure 48(b) and (d) the propagation times for through-resistance and ground-resistance defects in the driver, and Figure 48(c) and (e) the propagation times for through- and ground-resistance defects in the receiver. The defect-free results are generally nicely clustered at faster propagation speeds (though there are a few outliers that did not manage to propagate the test signal within the simulation period). The resistive defect results are more spread out, indicating that these faults would be detectable with our methodology. Unfortunately, the defect-free and defect-present propagation distributions overlap heavily. The implication is that a single test frequency will not suffice in order to achieve a high fault coverage. Instead, a set of different test frequencies will have to be used, based on analysis of each 3D via set, to increase the fault coverage. The relationship between test cost and fault coverage is a detailed optimization problem that we leave to future work.

Table 11 summarizes the propagation time distributions. Since the circuit parameters

were varied exponentially, the mean and standard deviation was calculated logarithmically. The mean propagation time for defect-free circuits is well within a single standard deviation for the mean propagation time of both resistive faults, highlighting the overlap noted in the graphs. Note however that the means for the resistive defects are substantially greater than for defect-free (76% and 420% greater for defects in the driver and in the receiver respectively). This suggests that simple integer clock-division may be sufficient for creating the set of test frequencies necessary to increase test coverage.

The results for the ground-resistance defect simulations (Figure 48(d) and (e)) are quite different from the through-resistance defect results. Notably, there are two widely-separated circuit responses regimes. To the left are the small-leakage faults, which the driving circuits are able to overcome fairly easy. To the right are the stuck-at-zero faults that simply cannot be charged over any reasonable length of time. This large variability is highlighted by the standard deviations (Table 11), which are an order of magnitude greater than those for the through-resistance defects[6]. The large response gap between these two fault types suggest that a design-for-yield (DFY) opportunity exists in addressing these faults. First the circuit designer would need to establish how large a ground-resistance defect is acceptable for the tier to still be considered good. Then, a DFY tool could tweak the 3D circuits place the switch-over point from small-leakage to stuck-at-ground slightly below that defect resistance. Our methodology would then be able to distinguish well between manageable and failure-inducing faults.

6.5 Physical Considerations

The discussion so far has been focused on the ideal case—*i.e.*, we have assumed a very low contact resistance (0.1Ω) between the probe tip and the 3D vias. In an actual manufacturing environment, low resistance cannot be guaranteed, so here we explore the effect of variable

[6]Because the propagation time is capped at 500ns due to simulation time constraints, the mean and standard deviations for ground-resistance defects are actually artificially fast. This effect is much less significant for the defect-free and through-resistance defect results because relatively few samples reach the cap there.

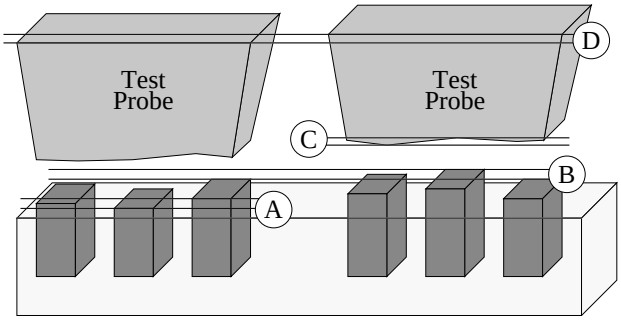


Figure 49. Sources of variation when using probe tips to form 3DV sets.

contact resistance on our proposed methodology.

Variation in the contact resistance can be caused both by process variation and by physical constraints. Relevant sources of variation are illustrated in Figure 49.[7] As labeled, these sources are (A) intra-set 3D via height variation, (B) inter-set 3D via height variation, (C) probe tip roughness, and (D) tip-to-tip height variation. Physical constraints are a result of the fine size of the 3D vias; a large probe force may damage these delicate structures, so a soft touch is required. Together, process variation and physical constraints significantly increase the realistic contact resistance.

Smith et al. [74] experimented with new probe cards designed to contact 3D vias. Specifically, they fabricated a MEMS-based probe card with a 40μm tip pitch. With this style probe card, they were able to achieve 1Ωcontact resistances in the general case and 10Ω contact resistance in the worst case (*i.e.*, with the lowest force and least over-travel). Unfortunately they did rely on scrub-marking to improve the contact quality, a technique which can not be employed in conjunction with our proposed technique. Therefore, we must anticipate larger contact resistances when probing multiple 3D vias at once.

To examine the impact of increasing the contact resistance, we performed another sensitivity analysis. Figure 50 shows the impact of increasing contact resistance with the driving 3D via on the propagation time. As the figure shows, our proposed technique is quite tolerant of a non-ideal contact resistance. Across the [1Ω–100Ω] range (which covers the

[7]The variation in Figure 49 has be exaggerated for clarity.

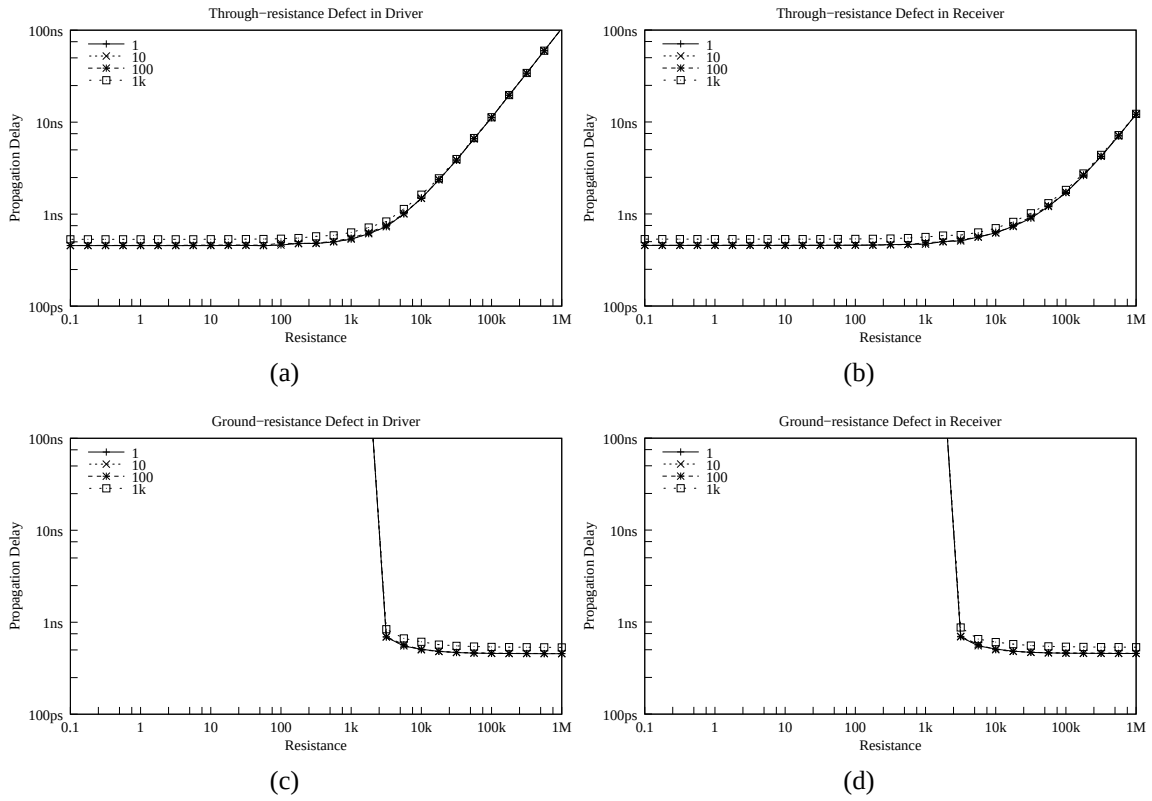


Figure 50. Propagation time results with increasing contact resistances to the driving 3D via.

expected contact resistances from [74]), the resistance has no impact on the propagation time. Beyond 100Ω, the contact resistance begins to have some small effect (the 1kΩ response is 80ps slower than the 100Ω response), but this a negligible impact.

Figure 51 also shows the impact of increasing contact resistance with the receiving 3D via. Once again, our technique proves very tolerant of non-ideal contact resistance; in this case, the effect of the larger contact resistance is not even visible in the plots. The difference between the 1Ω contact and the 1kΩ contact is less than 4% in the worst case.

Figure 52 summarizes the knee resistances (Figure 52(a)) and turn-on resistances (Figure 52(b)) across the [1Ω–1kΩ] contact resistance range for the driver (*i.e.*, when the probe makes poor contact with the driving 3D via); Figure 53 reports the same data for contact with the receiving via. These results confirm those in Figure 50 and Figure 51; even at a contact resistance well above the expected value, the contact resistance has only a minimal

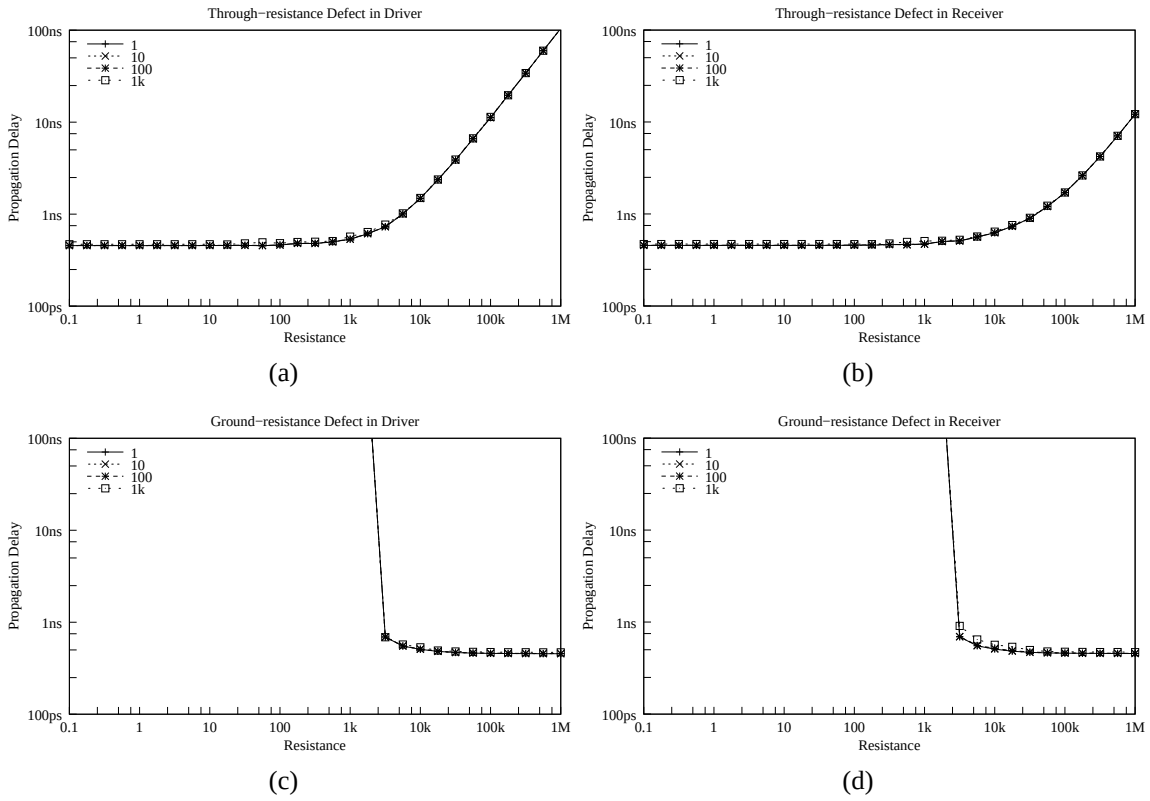
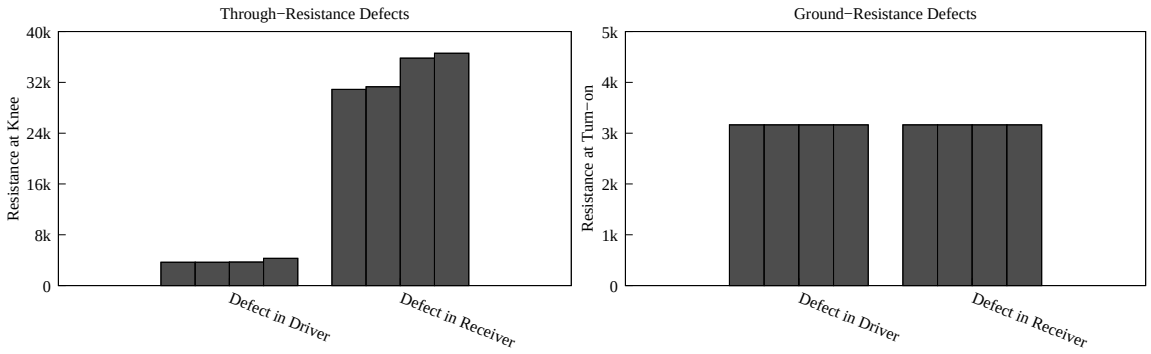


Figure 51. Propagation time results with increasing contact resistances to the receiving 3D via..

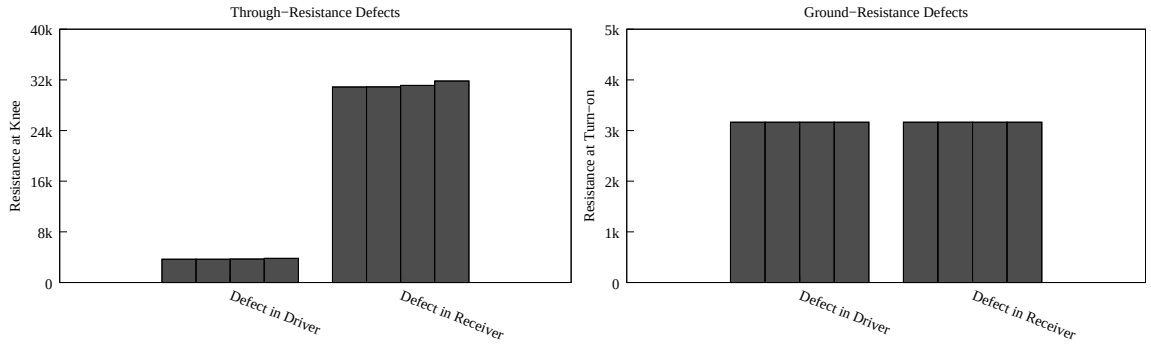
impact on circuit response. For ground-resistance defects, the contact resistance has no effect at all in the range of interest. It is important to note that the knee resistances are an order of magnitude greater for faults in the receiving 3D via. This is consistent with the pattern seen in Figure 45 for the other circuit parameters, as is expected. This reaffirms the observation that, because of the relative location of the probe tip capacitance, small-delay faults in the receiving 3D via are much harder to detect than those in the driving via.

This is not to say that poor contact quality does not have an impact. Comparing Figure 52 to the previous analysis reported in Figure 45, we can see that the knee and turn-on resistances for a poor contact are approximately the same as for the other circuit parameters. Unfortunately, this means the contact resistance does affect the fault detection capabilities of our methodology. The delay time associated with the contact resistance will add together and mask otherwise-detectable small-delay faults with resistances just beyond the knee and



(a) (b)

Figure 52. Knee and turn-on results with increasing contact resistances to the driving 3D via.



(a) (b)

Figure 53. Knee and turn-on results with increasing contact resistances to the receiving 3D via.

turn-on values. Fortunately though, the contact resistance does not completely overwhelm the rest of the circuit either. This means that we can still detect the large-delay and stuck-at faults that may be afflicting the 3D vias. This is key; even assuming a contact resistance orders of magnitude greater than expected, its effect remains insignificant enough to allow our methodology to effectively detect severe 3D via faults.

6.6 Summary

The ability to test 3D vias pre-bond in a high-volume manufacturing environment is one of the last significant roadblocks to industry's adoption of 3D integration technology. We have presented a new test methodology wherein traditional test probes are used to connect sets

of 3D vias together, forming new test paths that are both controllable and observable by traditional on-die test mechanisms. We have investigated some of the DFT constraints—3D via assignments and driving circuit tri-stating—necessary to make an unbonded 3D tier testable with our methodology. Finally, we have evaluated the feasibility of our methodology by modeling the test paths created and investigating their effectiveness at detecting the faults created by 3D via defects. Our simulations show that the presence of a fault alters the circuit response in a significant, observable manner, in spite of the significant load posed by the probe tip. High resistance (stuck-open) and high-leakage (stuck-at-zero) faults are easily detected, while small-delay and small-leakage faults more difficult but still detectable in some cases. Importantly, our investigation has identified several DFT opportunities for increasing the observability of these faults.

CHAPTER 7

RECENT DEVELOPMENTS

The 3D-aware test architecture presented in Chapter 3 was originally published in the International Test Conference in 2007. It was the first ever paper to propose a solution to the pre-bond test problem. Since that time, a vibrant research field has been created by the efforts of both academic and industrial research groups, of which the work presented in the previous chapters is just a small part. In this chapter, we will discuss some of recent results not previously discussed in this book that have been presented by others in the 3D test field.

7.1 IEEE P1838 Standard

Perhaps the most significant example of progress in the field of 3D-aware DFT is the proposed IEEE 1838 standard—*Standard for Test Access Architecture for Three-Dimensional Stacked Integrated Circuits* [2]. It is fundamentally an extension of the IEEE 1500 standard [4] for test wrappers to 3D. The proposed 3D-aware features are essentially identical to those we recommended in Chapter 3. These features have been further refined and detailed in subsequent works [48, 50].

The standard is composed of two elements, a set of tier-level wrapper features and a description language for specifying the wrapper design. The description language is a basic extension of the Boundary Description Language (BDL) defined in the IEEE 1149.1 standard [3]. More interesting is the wrapper specification. A key part of the 1500 standard is the specification of the wrapper cells that must be placed on every functional input and output of the module-under-test (MUT) (as discussed in Chapter 5). These wrapper cells serve two key functions: they enable the MUT to be tested independently of all logic external to the module, and they enable the interconnects between two modules to be tested independent of the internals of the modules.

The P1838 takes this concept and extends it to 3D tiers by adding the concepts of up and

down directionality to the wrapper. A 1500 wrapper has only inputs and outputs; a P1838 wrapper has inputs coming from up and from down neighboring tiers as well as outputs going to up and to down neighbors. Within this convention, the off-stack connections are defined to be on the bottom-most tier. A P1838 wrapper then has two important new modes of operation: *elevate* and *turn*. In the elevate mode, test data received from the down neighbor is passed to the up neighbor and vice versa. In the turn mode, test data from the down neighbor is returned to the down neighbor. To test a specific tier in the stack then, the target tier is placed in the turn mode so that its test responses are sent to the off-chip interface. All tiers below the test target are placed in the elevate mode to pass test data along between the off-chip interface and the target. In the parlance of Chapter 3, the P1838 is the LTC.

The P1838 is compatible with the 1500 and 1149.1 standards, so an example 3D stack might have an 1149.1 wrapper around the entire stack, P1838 wrappers around each tier, and 1500 wrappers around the individual modules within each tier. This is equivalent to the CTC—LTC—ITC hierarchy described in Chapter 3.

It is important to note that the P1838 is a robust design. Though targeted specifically to TSV-based 3D ICs, it can also be applied to other variations of the 3D theme, like wire-bound 3D stacks and 2.5D designs built on interposers [19]. This flexibility makes P1838 a very powerful standard for empowering the 3D industry.

7.2 Pre-bond Test

The works presented in this book have focused mostly on pre-bond test of the circuits internal to each tier. However, this is just one facet of the pre-bond test challenge. Here we explore the recent contributions of other groups to the field of pre-bond test.

The authors of [22] propose another 3D test architecture for enabling pre-bond and post-bond test of 3D ICs, very similar to both Chapter 3 and the P1838 standard. In this work, they focus on explicitly 1149.1 and 1500 standards compatibility, relying on just

the minimum four test signals defined in those standards to enable 3D test. With their wrapper designs, they achieve 3D test with only an insignificant 0.15% area overhead. This highlights the power of scan-based testing to provide excellent coverage at low cost.

In [20], the authors assume the basic test architecture described in Chapter 3 but extend it to multiple towers. In this work, they aim for ultimate flexibility, considering not just multiple chip stacks on a single interposer but also so-called sub-stacks of chips on top of larger stacks. This work really highlights the flexibility of our basic test architecture and is one of the building blocks of the P1838 standard.

In [62], the authors study the design of the buried probe pad arrays that are necessary for pre-bond test. Specifically, they look at the co-design of the scan chain end pads and the power-delivery probe pads. With the former, more pads increase test access and so decrease test cost, but consume more area, limiting the number of pads available for power delivery and the area available for 3D vias. With the latter, more pads increase the power delivery quality but limit the test access and again the available 3D via area. The authors propose CAD algorithms for optimizing this design problem, identifying optimal trade-off points between test access and power delivery to the pre-bond tier under test.

The work presented in [55] is closely related to that in [56], the work that proposed probing 3DV sets with active probe cards. In this new work, they improve on their fault resolution capability by using multiple test insertions to help resolve fault locations. Simply put, if a 3D via is probed in two different sets, and only one of those sets fails, that 3D via is known to be good. This process-of-elimination can be extended to all 3D vias, allowing faulty vias to be precisely determined. The downside is increased cost of the many test insertions.

7.3 Post-bond Test

While the works in this book have focused on pre-bond test, partial-stack and post-bond test do create some new challenges and opportunities that are not found in traditional ICs.

In [57] and [59], the authors analyzed how the stacking order in the 3D IC affected the total test cost. This is yet another variable for designers to consider, in addition to more basic concerns like IR-drop, thermal dissipation, routing cost, and die size. Using the "bottom chip" convention of the P1838 standard, test data that is hoisted to the top chip must scan through every chip below it. That makes this test data more expensive to transport than data going to the bottom chip. Therefore, it is cheaper to sort the 3D stack in terms of increasing test complexity so that test data traverses a minimum number of tiers. The authors also consider other factors, such as multiple test applications in partial-stack test and limited 3D via resources for test. The authors conclude that optimizing just for post-bond test can significantly increase overall test cost, demanding a more thoughtful design of the stack's test architecture.

In [30], the authors propose a new test-specific logical organization for the 3D vias in the stack to optimize the test cost. Independent of the 3D vias' functional purposes, they are organized into an addressable array for testing purposes. The authors then use MBIST-based test structures to activate and test the 3D vias. Utilizing this scheme, they report 85.2% and 93.6% reductions in area overhead and test time respectively as compared to a simple 1500-based test method. They reduce the area by not dedicating a boundary cell to each 3D via, and they reduce test time by using BIST, rather than scanning every test pattern in from the ATE.

In [18], the authors tackle the problem of 3D wrapper design. They note that while test time can be reduced by designing 3D wrapper chains, using a large number of 3D vias to create these chains can create routing and congestion problems. They propose a new heuristic algorithm for designing 3D wrappers that takes advantage of 3D design while minimizing 3D via usage. They report a 33% reduction in 3D via utilization compared to prior schemes.

7.4 3D Assembly

Testing 3D chips pre-bond is critical to the economic viability of the fledgling 3D IC industry. However, it is not the end of the story. With the pass/fail data in hand, manufacturers must use this data to increase the yield of the final chip stacks by minimizing the number of good tiers that get bound to failed tiers. Manufactures have two choices for actually stacking 3D ICs, wafer-bonding and chip-bonding. In chip bonding, the chips are diced out from the wafers, then bond into the chip stacks. This allows only known-good chips to be bound together, but the small size and large quantity of chips makes handling difficult. In wafer bonding, wafers are bound together, then the stacks are diced out. Handling is then much easier, but bonding some good chips to known-failed chips is unavoidable. There is a third option, chip-to-wafer bonding, which has similar trade-offs to chip bonding.

However, it is still possible to optimize the number of known-good chip stacks, even when wafer-bonding is used. In [79], they propose matching algorithms for selecting wafers to bond together to maximize the number of good stacks that are produced. They examine a large variety of factors, including stack height, chip size, chip yield, and repository size (the number of wafers from which the bonding pair may be chosen). They consider both replenished and non-replenished repositories, and they consider different optimization goals (*e.g.*, maximizing the number of good-good stacks versus maximizing the number of fail-fail stacks). By utilizing their matching algorithms, they are able to improve the final stack yields by as much as 13.4%.

In [72], the authors propose a novel new approach to packing chips onto a wafer. Rather than simply repeating the chip design across the entire wafer, they divide the wafer into four quadrants. The chips in each quadrant rotated $\pm 90^o$ with respect to the adjacent quadrants. This provides a significant advantage when wafer bonding. In basic wafer bonding, there is only one possible orientation for a wafer when attempting to maximize the final yield; with the quadrant system, there are four orientations which effectively quadruples the wafer repository size. The greatly increasing the number of potential wafer pairs, improving the

chance that near-perfect matches can be made. The authors report a 25% improvement in yield utilizing this technique.

In [26] and [78], the authors present the novel idea of stacking multiple, redundant tiers in the case that some fail or that the wafer-bond process is too inflexible to produce good stacks with the minimum number of tiers. Basically, as long as a tier can pass inter-tier signals along, the rest of the tier can be faulty without failing the stack. This is particularly applicable to stacked memories, where the memory can still work at a reduced capacity due to a faulty tier so long as that tier does not disable the memory bus. The authors report a 59% in stack yield when applying this technique in conjunction with wafer matching.

7.5 3D Via Repair

Even if a manufacturer is able to optimally select two known-good tiers to bond together, the resulting stack is not guaranteed good. The bonding process is subject to faults just like any other process. To attempt to recover from a failed bond, many researchers have looked into methods for repairing or replacing faulty 3D vias.

In [29], the authors proposed a redundancy scheme to allow faulty 3D vias to be replaced with good 3D vias post-bond. They accomplish this by subdividing the 3D vias into repairable ordered-sets composed of N functional 3D vias and one redundant via. If a via fails within the set, the signals in the set shift one 3D via over via multiplexers. This allows each via set to recover from one failed 3D via. Using this simple design, the authors claim they can recover enough failed 3D vias to ensure 99.99% bond quality between tiers. Of course, this method has implications for the timing across the 3D interface since the circuit designer does not know if the signal will end up taking the primary or back-up path; this uncertainly must be accounted for in the design margin.

In [87], the authors provide an in-depth investigation into the trade-off between 3D via failure rate and redundancy costs. Whereas the previous work just assumes one redundant 3D via per set, this work varies the number of redundant 3D vias to optimally match the

failure rate and balance the cost of repair against the gain in final stack yield. Assuming the failure rate is a well-established value for a given 3D process, the authors claim they can achieve 100% yield for a small cost.

The previous works assumes a uniform distribution of 3D via faults. The authors in [34] assert that this is incorrect; process analysis in fact shows that 3D via faults tend to be spatially correlated (*i.e.*, if there are two faulty 3D vias, there is a high likelihood that they are located near one another). They suggest an update to the 3D via repair scheme that spaces out the 3D vias in the sets to counteract this correlation. The authors claim a significant improvement in repair capability in the face of grouped faulty 3D vias. The downside is that both the signaling margins and the repair overhead are penalized by this additional capability.

CHAPTER 8

CONCLUSION

In this dissertation we have proposed several DFT techniques specific to 3D stacked IC systems. The goal has explicitly been to create techniques that integrate easily with existing IC test systems. Specifically, this has meant utilizing scan- and wrapper-based techniques because these are the foundations of the digital IC test industry.

First, we described a general test architecture for 3D ICs. In this architecture, each tier of a 3D design is defined to be an independently-testable block. The tier is then wrapped in test control logic that both manages tier test pre-bond and integrates the tier into the large test architecture post-bond. To enable pre-bond test of all the circuits internal to the tier, we described a new kind of boundary scan wherein each 3D via is supplemented with DFT logic to provide the necessary test control and observation. Our experimental results showed that this boundary scan technique could be implemented in a block-partitioned 3D design with a negligible overhead. To ensure the operation of the test hardcore, we proposed a new design methodology for these nets that ensures both pre-bond functionality and post-bond optimality. We showed how all these design techniques were utilized in the development of the 3D-MAPS test vehicle, which has proven their effectiveness.

Second, we extended these DFT techniques to circuit-partitioned designs. We found that the boundary scan design is low enough overhead to meet the test and cost requirements of all but the the most tightly integrated 3D designs. We examined the case of the 3D port-split register file, a design for which pre-bond boundary scan was insufficient. We presented a new 3D-aware MBIST technique that could be used in conjunction with our pre-bond test architecture to fully verify the register file while avoiding the problems of 3D boundary scan. Most significantly, the combination of 3D design and the new MBIST algorithm reduced the cost of test by nearly 40%, demonstrating that test cost reduction is another potential benefit of 3D integration, in addition to speed, power, area, and routability

benefits.

Third, we examined the design of test wrappers for 3D IP, a special case of 3D test logic where the 3D stack designer cannot know the design of the individual IP blocks. Producing 3D wrappers required a new algorithm because existing techniques only produced a single wrapper, not the several pre-bond wrapper and single post-bond wrapper demanded by a 3D system. Our algorithm, based off the BFD sorting and KL partitioning algorithms, succeeded in producing 3D wrappers that minimized both test time and design cost.

Finally, we looked at the 3D vias themselves to develop a low-cost, high-volume pre-bond test methodology appropriate for production-level test. We described the shorting probes methodology, wherein large test probes are used to contact multiple small 3D vias. This technique has the notable benefits of being an all-digital test method and of integrating seamlessly into existing test flows. Our experimental results demonstrated two key facts: neither the large capacitance of the probe tips nor the process variation in the 3D vias and the probe tips significantly hinders the testability of the circuits. Thus we showed shorting probes to be an effective method for detecting stuck-at and stuck-open faults in unbonded 3D tiers.

Taken together, this body of work has defined a complete test methodology for testing 3D ICs pre-bond, eliminating one of the key hurdles to the commercialization of 3D technology by the IC industry. We look forward to seeing the continued adoption of these designs by the industry and the incredible new products that result.

REFERENCES

[1] "Magic VLSI layout tool." http://opencircuitdesign.com/magic/release.html.

[2] "Test access architecture for three-dimensional stacked integrated circuits," *Proposed IEEE Standard P1838*.

[3] "Test access port and boundary-scan architecture," *IEEE Standard 1149.1*, 1990.

[4] "Testability method for embedded core-based integrated circuits," *IEEE Standard 1500*, 2005.

[5] "JSED229: Wide I/O single data rate," *JEDEC Standard*, December 2011.

[6] Abadir, M. S. and Reghbati, H. K., "Functional testing of semiconductor random access memories," *Computing Surveys*, vol. 15, pp. 175–198, September 1983.

[7] ASU, "Predictive technology model." http://ptm.asu.edu/, 2011.

[8] Berthelot, D., Chaudhuri, S., and Savoj, H., "An efficient linear time algorithm for scan chain optimization and repartitioning," in *Proceedings of the International Test Conference*, pp. 781–787, October 2002.

[9] Bhansali, S., Chapmann, G., Friedman, E., Ismail, Y., Mukund, P., Tebbe, D., and Jain, V., "3-D heterogeneous sensor system on a chip for defense and security applications," in *Proceedings of SPIE*, vol. 5417, pp. 413–424, 2004.

[10] Bhavsar, D. K. and Davies, R. A., "Scan islands—a scan partitioning architecture and its implementation on the Alpha 21364 processor," in *Proceedings of the VLSI Test Symposium*, pp. 16–21, April 2002.

[11] Black, B., Annavaram, M., Brekelbaum, N., DeVale, J., Jiang, L., Loh, G. H., McCauley, D., Morrow, P., Nelson, D. W., Pantuso, D., Reed, P., Rupley, J., Shankar, S., Shen, J., and Webb, C., "Die stacking (3D) microarchitecture," in *Proceedings of the International Symposium on Microarchitecture*, pp. 469–479, December 2006.

[12] Bonhomme, Y., Girard, P., Guiller, L., Landrault, C., and Pravossoudovitch, S., "Efficient scan chain design for power minimization during scan testing under routing constraint," in *Proceedings of the International Test Conference*, pp. 488–493, September 2003.

[13] Breuer, M. A. and Friedman, A. D., *Diagnosis and Reliable Design of Digital Systems*. Computer Science Press, Incorporated, 1976.

[14] Cadence, "http://www.cadence.com/us/pages/default.aspx," 2010.

[15] Cascade Microtech, "http://www.cmicro.com/." 2011.

[16] Chen, P.-Y., Wu, C.-W., and Kwai, D.-M., "On-chip testing of blind and open-sleeve TSVs for 3D IC before bonding," in *Proceedings of the VLSI Test Symposium*, pp. 263–268, April 2010.

[17] Chen, P.-Y., Wu, C.-W., and Kwai, D.-M., "On-chip TSV testing for 3D IC before bonding using sense amplification," in *Proceedings of the Asian Test Symposium*, pp. 450–455, November 2009.

[18] Cheng, Y., Zhang, L., Han, Y., Liu, J., and Li, X., "Wrapper chain design for testing TSVs minimization in circuit-partitioned 3D SoC," in *Proceedings of the Asian Test Symposium*, pp. 181–186, November 2011.

[19] Chi, C.-C., Marinissen, E. J., Goel, S. K., and Wu, C.-W., "Post-bond testing of 2.5D-SICs and 3D-SICs containing a passive silicon interposer base," in *Proceedings of the International Test Conference*, pp. 1–10, September 2011.

[20] Chi, C.-C., Marinissen, E. J., Goel, S. J., and Wu, C.-W., "DfT architecture for 3D-SICs with multiple towers," in *Proceedings of the European Test Symposium*, pp. 51–56, May 2011.

[21] Cho, M., Liu, C., Kim, D. H., Lim, S. K., and Mukhopadhyay, S., "Design method and test structure to characterize and repair TSV defect induced signal degradation in 3D system," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 694–697, November 2010.

[22] Chou, C.-W., Li, J.-F., Chen, J.-J., Kwai, D.-M., Chou, Y.-F., and Wu, C.-W., "A test integration methodology for 3D integrated circuits," in *Proceedings of the Asian Test Symposium*, pp. 377–382, December 2010.

[23] Cong, J. and Lim, S. K., "Multiway partitioning with pairwise movement," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 512–516, November 1998.

[24] Das, S., Chandrakasan, A., and Reif, R., "Design tools for 3-D integrated circuits," in *Proceedings of the Asia South Pacific Design Automation Conference*, pp. 53–56, January 2003.

[25] GLOBALFOUNDARIES, "http://www.globalfoundries.com/," 2010.

[26] Hamdioui, S. and Taouil, M., "Yield improvement and test cost optimization for 3D stacked ICs," in *Proceedings of the Asian Test Symposium*, pp. 480–485, November 2011.

[27] Healy, M. B., Athikulwongse, K., Goel, R., Hossain, M. M., Kim, D. H., Lee, Y.-J., Lewis, D. L., Lin, T.-W., Liu, C., Jung, M., Ouellette, B., Pathak, M., Sane, H., Shen, G., Woo, D. H., Zhao, X., Loh, G. H., Lee, H.-H. S., and Lim, S. K., "Design and analysis of 3D-MAPS: A many-core 3D processor with stacked memory," in *Proceedings of the Custom Integrated Circuits Conference*, pp. 1–4, September 2010.

[28] Hirech, M., Beausang, J., and Gu, X., "A new approach to scan chain reordering using physical design information," in *Proceedings of the International Test Conference*, pp. 348–355, October 1998.

[29] Hsieh, A.-C., Hwang, T., Chang, M.-T., Tsai, M.-H., Tseng, C.-M., and Li, H.-C., "TSV redundancy: Architecture and design issues in 3D IC," in *Proceedings of the Design, Automation, and Test in Europe Conference*, pp. 166–171, March 2010.

[30] Huang, Y.-J., Li, J.-F., Chen, J.-J., Kwai, D.-M., Chou, Y.-F., and Wu, C.-W., "A built-in self-test scheme for the post-bond test of TSVs in 3D ICs," in *Proceedings of the VLSI Test Symposium*, pp. 20–25, May 2011.

[31] Iyengar, V., Chakrabarty, K., and Marinissen, E. J., "Test wrapper and test access mechanism co-optimization for system-on-chip," *Journal of Electronic Testing: Theory and Applications*, vol. 18, no. 2, pp. 213–230, 2002.

[32] Jiang, L., Huang, L., and Xu, Q., "Test architecture design and optimization for three-dimensional SOCs," in *Proceedings of the Design, Automation, and Test in Europe Conference*, pp. 220–225, April 2009.

[33] Jiang, L., Xu, Q., Chakrabarty, K., and Mak, T. M., "Layout-driven test-architecture design and optimization for 3D SOCs under pre-bond test-pin-count constraint," in *Proceedings of the International Conference on Computer-Aided Design*, November 2009.

[34] Jiang, L., Xu, Q., and Eklow, B., "On effective TSV repair for 3D-stacked ICs," in *Proceedings of the Design, Automation, and Test in Europe Conference*, pp. 793–798, March 2012.

[35] Katti, G., Stucchi, M., Meyer, K. D., and Dehaene, W., "Electrical modeling and characterization of through silicon via for three-dimensional ICs," *IEEE Transactions on Electron Devices*, vol. 57, no. 1, pp. 256–262, 2010.

[36] Kim, D. H., Athikulwongse, K., Healy, M., Hossain, M., Jung, M., Khorosh, I., Kumar, G., Lee, Y.-J., Lewis, D., Lin, T.-W., Liu, C., Panth, S., Pathak, M., Ren, M., Shen, G., Song, T., Woo, D. H., Zhao, X., Kim, J., Choi, H., Loh, G., Lee, H.-H., and Lim, S. K., "3D-MAPS: 3D massively parallel processor with stacked memory," in *Digest of Technical Papers of the International Solid-State Circuits Conference*, pp. 188–189, February 2012.

[37] Kim, D. H., Mukhopadhyay, S., and Lim, S. K., "Fast and accurate analytical modeling of through-silicon-via capacitive coupling," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 1, no. 2, pp. 168–180, 2011.

[38] Knaizuk, J. and Hartmann, C. R. P., "An optimal algorithm for testing stuck-at faults in random access memories," *IEEE Transactions on Computers*, vol. 100, no. 11, pp. 1141–1144, 1977.

[39] Konemann, B., Mucha, J., and Zwiehoff, G., "Built-in logic block observation techniques," in *Proceedings of the International Test Conference*, pp. 1–6, October 1979.

[40] Lewis, D. L. and Lee, H.-H. S., "A scan-island based design enabling pre-bond testability in die-stacked microprocessors," in *Proceedings of the International Test Conference*, pp. 1–8, October 2007.

[41] Lewis, D. L. and Lee, H.-H. S., "Testing circuit-partitioned 3D IC designs," in *Proceedings of the International Symposium on VLSI*, pp. 139–144, May 2009.

[42] Lewis, D. L., Panth, S., Zhao, X., Lim, S. K., and Lee, H.-H. S., "Designing 3D test wrapper for pre-bond and post-bond test of 3D embedded cores," in *Proceedings of the International Conference on Computer Design*, pp. 90–95, October 2011.

[43] Li, F., Nicopoulos, C., Richardson, T., Xie, Y., Vijaykrishnan, N., and Kandemir, M., "Design and management of 3D chip multiprocessors using network-in-memory," in *Proceedings of the International Symposium on Computer Architecture*, pp. 130–141, June 2006.

[44] Lim, S. K., *Practical Problems in VLSI Physical Design Automation*. Springer Verlag, 2008.

[45] Lo, C.-Y., Hsing, Y.-T., Denq, L.-M., and Wu, C.-W., "SOC test architecture and method for 3-D ICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 10, pp. 1645–1649, 2010.

[46] Lou, Y., Yan, Z., Zhang, F., and Franzon, P. D., "Comparing through-silicon-via (TSV) void/pinhole defect self-test methods," in *Proceedings of the International Workshop on Testing Three-Dimensional Stacked Integrated Circuits*, pp. 1–7, November 2010.

[47] Makar, S., "A layout-based approach for ordering scan chain flip-flops," in *Proceedings of the International Test Conference*, pp. 341–347, October 1998.

[48] Marinissen, E. J., Chi, C.-C., Verbree, J., and Konijnenburg, M., "3D DFT architecture for pre-bond and post-bond testing," in *Proceedings of the International 3D System Integration Conference*, pp. 1–8, November 2010.

[49] Marinissen, E. J., Goel, S. K., and Lousberg, M., "Wrapper design for embedded core test," in *Proceedings of the International Test Conference*, pp. 911–920, October 2000.

[50] Marinissen, E. J., Verbree, J., and Konijnenburg, M., "A structured and scalable test access architecture for TSV-based 3D stacked ICs," in *Proceedings of the VLSI Test Symposium*, pp. 269–274, April 2010.

[51] McLaurin, T., "The challenge of testing the ARM Cortex-A8^{TM} microprocessor core," in *Proceedings of the International Test Conference*, pp. 1–10, October 2006.

[52] Mentor Graphics, "http://www.mentor.com," 2007.

[53] Minz, J., Zhao, X., and Lim, S. K., "Buffered clock tree synthesis for 3D ICs under thermal variations," in *Proceedings of the Asia South Pacific Design Automation Conference*, pp. 504–509, March 2008.

[54] Mysore, S., Agrawal, B., Srivastava, N., Lin, S.-C., Banerjee, K., and Sherwood, T., "Introspective 3D chips," in *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 264–273, October 2006.

[55] Noia, B. and Chakrabarty, K., "Identification of defective TSVs in pre-bond testing of 3D ICs," in *Proceedings of the Asian Test Symposium*, pp. 187–194, November 2011.

[56] Noia, B. and Chakrabarty, K., "Pre-bond probing of TSVs in 3D stacked ICs," in *Proceedings of the International Test Conference*, pp. 1–9, September 2011.

[57] Noia, B., Chakrabarty, K., and Marinissen, E. J., "Optimization methods for post-bond die-internal/external testing in 3D stacked ICs," in *Proceedings of the International Test Conference*, pp. 1–9, November 2010.

[58] Noia, B., Chakrabarty, K., and Xie, Y., "Test-wrapper optimization for embedded cores in TSV-based three-dimensional SOCs," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 70–77, November 2009.

[59] Noia, B., Goel, S. K., Chakrabarty, K., Marinissen, E. J., and Verbree, J., "Test-architecture optimization for TSV-based 3D stacked ICs," in *Proceedings of the European Test Symposium*, pp. 24–29, May 2010.

[60] Nunomura, Y. and Manijikian, N., "M32R/D-integrating DRAM and microprocessor," *IEEE MICRO*, vol. 17, no. 6, pp. 40–48, 1997.

[61] OpenCores, "http://opencores.org." January 2011.

[62] Panth, S. and Lim, S. K., "Scan chain and power delivery network synthesis for pre-bond test of 3D ICs," in *Proceedings of the VLSI Test Symposium*, pp. 26–31, May 2011.

[63] Parvathala, P., Maneparambil, K., and Lindsay, W., "FRITS—a microprocessor functional BIST method," in *Proceedings of the International Test Conference*, pp. 590–598, October 2002.

[64] Patti, B., "Keynote address: Testing in a new dimension," in *IEEE International Workshop on Testing Three-Dimensional Stacked Integrated Circuits*, November 2010.

[65] Pavlidis, V. and Friedman, E., "3-D topologies for networks-on-chip," in *Proceedings of the International SOC Conference*, pp. 285–288, September 2006.

[66] Puttaswamy, K. and Loh, G. H., "Implementing caches in a 3D technology for high performance processors," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 525–532, October 2005.

[67] Puttaswamy, K. and Loh, G. H., "Dynamic instruction schedulers in a 3-dimensional integration technology," in *Proceedings of the Great Lakes Symposium on VLSI*, pp. 153–158, April 2006.

[68] Puttaswamy, K. and Loh, G. H., "The impact of 3-dimensional integration on the design of arithmetic units," in *Proceedings of the International Symposium on Circuits and Systems*, pp. 4951–4954, May 2006.

[69] Puttaswamy, K. and Loh, G. H., "Implementing register files for high-performance microprocessors in a die-stacked (3D) technology," in *Proceedings of the International Symposium on VLSI*, pp. 384–389, March 2006.

[70] Riley, M., Bushard, L., Chelstrom, N., Kiryu, N., and Ferguson, S., "Testability features of the first-generation Cell processor," in *Proceedings of the International Test Conference*, pp. 111–119, November 2005.

[71] Samsung, "http://www.samsung.com/presscenter/pressrelease/pressrelease.asp?seq=20060413_0000246668." 2006.

[72] Singh, E., "Exploiting rotational symmetries for improved stacked yields in W2W 3D-SICs," in *Proceedings of the VLSI Test Symposium*, pp. 32–37, May 2011.

[73] Smith, K., Hanaway, P., Jolley, M., Gleason, R., Fournier, C., and Strid, E., "KGD probing of TSVs at 40μm array pitch," in *Proceedings of the International Workshop on Testing Three-Dimensional Stacked Integrated Circuits*, pp. 1–7, November 2010.

[74] Smith, K., Hanaway, P., Jolley, M., Gleason, R., Strid, E., Daenen, T., Dupas, L., Knuts, B., Marinissen, E. J., and Dievel, M. V., "Evaluation of TSV and micro-bump probing for wide I/O testing," in *Proceedings of the International Test Conference*, pp. 1–10, September 2011.

[75] Stucchi, M., Perry, D., Katti, G., and Dehaene, W., "Test structures for characterization of through silicon vias," in *Proceedings of the Internation Conference on Microelectronic Test Structures*, pp. 130–134, March 2010.

[76] Suk, D. S. and Reddy, S. M., "A march test for functional faults in semiconductor random access memories," *IEEE Transactions on Computers*, vol. C-30, no. 12, pp. 982–985, 1981.

[77] Tan, P., Le, T., Ng, K.-H., Mantri, P., and Westfall, J., "Testing of UltraSPARC T1 microprocessor and its challenges," in *Proceedings of the International Test Conference*, pp. 1–10, October 2006.

[78] Taouil, M. and Hamdioui, S., "Layer redundancy based yield improvement for 3D wafer-to-wafer stacked memories," in *Proceedings of the European Test Symposium*, pp. 45–50, May 2011.

[79] Taouil, M., Hamdioui, S., Verbree, J., and Marinissen, E. J., "On maximizing the compound yield for 3D wafer-to-wafer stacked ICs," in *Proceedings of the International Test Conference*, pp. 1–10, November 2010.

[80] Tsai, M., Klooz, A., Leonard, A., Appel, J., and Franzon, P., "Through silicon via (TSV) defect/pinhole self test circuit for 3D-IC," in *Proceedings of the International Conference on 3D System Integration*, pp. 1–8, September 2009.

[81] Weeden, O., "Probe card tutorial," 2003. http://www.keithley.com/data?asset=13263.

[82] Wong, E. and Lim, S. K., "3D floorplanning with thermal vias," in *Proceedings of the Design, Automation, and Test in Europe Conference*, pp. 878–883, March 2006.

[83] Woo, D. H., Seong, N. H., Lewis, D. L., and Lee, H.-H. S., "An optimized 3D-stacked memory architecture by exploring excessive, high-density TSV bandwidth," in *Proceedings of the International Symposium on High-Performance Computer Architecture*, pp. 1–12, January 2010.

[84] Wu, X., Chen, Y., Chakrabarty, K., and Xie, Y., "Test-access mechanism optimization for core-based three-dimensional SOCs," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 212–218, November 2009.

[85] Zhao, X., Lewis, D. L., Lee, H.-H. S., and Lim, S. K., "Pre-bond testable low-power clock tree design for 3D stacked ICs," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 191–196, November 2009.

[86] Zhao, X., Lewis, D. L., Lee, H.-H. S., and Lim, S. K., "Low-power clock tree design for pre-bond testing of 3D stacked ICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, pp. 732–745, May 2011.

[87] Zhao, Y., Khursheed, S., and Al-Hashimi, B. M., "Cost-effective TSV grouping for yield improvement of 3D-ICs," in *Proceedings of the Asian Test Symposium*, pp. 201–206, November 2011.