PHYSICAL DESIGN METHODOLOGIES FOR MONOLITHIC 3D ICS

A Dissertation
Presented to
The Academic Faculty

by

Shreepad Amar Panth

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

Georgia Institute of Technology
May 2015

Copyright © 2015 by Shreepad Amar Panth

PHYSICAL DESIGN METHODOLOGIES FOR MONOLITHIC 3D ICS

Approved by:

Dr. Sung Kyu Lim, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Saibal Mukhopadhyay
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Arijit Raychowdhury
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Azad Naeemi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Hyesoon Kim
College of Computing
Georgia Institute of Technology

Date Approved: March 13, 2015

ACKNOWLEDGEMENTS

Finishing my Ph.D. has been a long journey, and it wouldn't have been possible without the assistance of many people. I would like to thank all those who helped me along the way.

Firstly, I would like to thank my advisor, Dr Sung Kyu Lim, for guiding me and shaping my research. He gave me the chance to pursue the highest academic degree possible, in one of the best research institutions in the world.

I would like to thank Dr. Saibal Mukhopadhyay and Dr. Arijit Raychowdhury for suggestions and guidance on my research. In addition, I thank Dr. Azad Naeemi and Dr. Hyesoon Kim for serving on my dissertation defence committee.

The bulk of my research was spent working under a project with Qualcomm, and I would like to thank Dr. Kambiz Samadi, Dr. Yang Du, and Pratyush Kamal for providing valuable feedback and an industrial viewpoint to guide my research.

I would like to thank the past and current members of the GTCAD lab: Dr. Michael Healy, Mohit Pathak, Dr. Dae Hyun Kim, Dr. Xin Zhao, Dr. Krit Athikulwongse, Dr. Young-Joon Lee, Dr. Moonong Jung, Taigon Song, Yarui Peng, Sandeep Samal, Neela Lohith, Kyung Wook Chang, Bon Woong Ku and Kartik Acharya for providing expertise in areas I was unfamiliar with, tools and scripts, as well their time for me to bounce ideas off of. I also thank David Webb and Keith May, from the school of ECE's IT department, for responding to hundreds of my requests over the years.

Lastly, I would like to thank my family – my parents, grandparents, and sister for their support not just through my Ph.D., but throughout my life.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . . **iii**

LIST OF TABLES . . . . **vii**

LIST OF FIGURES . . . . **ix**

SUMMARY . . . . **xiv**

I INTRODUCTION **1**

1.1 Overview of Monolithic 3D ICs 2

1.1.1 Fabrication Techniques 2

1.1.2 Design Styles 4

1.2 Organization and Contributions 5

II DESIGN-FOR-TEST FOR TSV-BASED 3D ICS **7**

2.1 Scan-Chain Design for 3D ICs 8

2.1.1 3D Scan Chain Construction 8

2.1.2 Reuse of Signal TSVs 10

2.1.3 Broken Scan Chains 10

2.1.4 Experimental Results 11

2.2 Transistion-delay-fault Testing for 3D ICs with IR-drop Study 13

2.2.1 Transition-delay-fault Architecture 14

2.2.2 Probe-pad Placement and PDN Design 17

2.2.3 Design and Analysis Flow 20

2.2.4 Experimental Results 21

2.3 Test-time Estimation for 3D ICs 29

2.3.1 Die-level partitioning 30

2.3.2 Block-level partitioning 35

2.3.3 Case Studies 42

2.4 Summary 47

III PHYSICAL DESIGN FOR BLOCK-LEVEL MONOLITHIC 3D ICS . . . 48
3.1 3D Floorplanning with Monolithic Inter-tier Vias 49
3.1.1 Problem Formulation and Overview 49
3.1.2 Floorplanning Engine 50
3.1.3 Post-Floorplan Refinement (PFPR) 51
3.1.4 MIV Planning Algorithm 52
3.2 Floorplan Quality Evaluation 55
3.2.1 Experimental Setup 55
3.2.2 Floorplanner Validation 57
3.2.3 Monolithic 3D vs. TSV-based 3D 58
3.3 Inter-Tier Performance Differences 58
3.3.1 Source of Inter-Tier Performance Differences 58
3.3.2 Degraded Interconnects on the Bottom Tier 60
3.3.3 Degraded Transistors on the Top Tier 61
3.4 Performance-Difference-Aware Design and Analysis Flow 63
3.4.1 Performance-Difference-Aware Floorplanner 64
3.4.2 Performance-Difference-Aware Analysis 66
3.5 Power-Performance Study 67
3.5.1 Identical Performance on Both Tiers 67
3.5.2 Impact of Inter-Tier Performance Differences 69
3.5.3 Overall Comparisons 73
3.5.4 Block Folding 73
3.6 Summary 75
IV PHYSICAL DESIGN FOR GATE-LEVEL MONOLITHIC 3D ICS 76
4.1 Congestion-Aware Placement for Gate-level Monolithic 3D ICs 77
4.1.1 Overall Design Flow 77
4.1.2 Monolithic 3D IC Placement 78
4.1.3 Routability-Driven Partitioning 82
4.1.4 Router-based 3D-Via Insertion 92

4.1.5 Experimental Results 93
4.1.6 Comparison with Existing 3D Placers 104
4.2 Monolithic 3D IC Design With Commercial 2D IC Tools 106
4.2.1 CAD Methodology 107
4.2.2 Power Benefit Study 112
4.3 IR-drop Aware Partitioning for Monolithic 3D ICs 117
4.3.1 Motivation and Objectives 118
4.3.2 Design and Analysis Flow 119
4.3.3 Experimental Results 124
4.4 Summary 132

V CONCLUSIONS AND FUTURE DIRECTIONS 133

REFERENCES 135

PUBLICATIONS 141

VITA 145

LIST OF TABLES

1 Statistics for different scan chain configurations. 12

2 Design Statistics for two designs, split by die. 23

3 Post-bond test time results. All test times are in cycles 24

4 The optimal test times (in cycles) achieved for a two-die circuit, along with the TSV usage at which this optimum time is reached. 32

5 The test times for die-level partitioning of a three-die 3D IC, considering both uniform and tapered TSV constraints. 32

6 The test times for die-level partitioning of a four-die 3D IC, considering both uniform and tapered TSV constraints. 33

7 Details of benchmark circuits used, showing the average and standard deviation of the test data volume among all modules. 43

8 Design Statistics for All Benchmarks 55

9 Comparison between the proposed floorplanner and Cadence Encounter. . . 57

10 A comparison of wirelength, timing and top net power of 2D versus 3D . . 59

11 Various interconnect parameters 61

12 The change in resistivity values of different metal layers in the Nangate 45nm library due to Tungsten interconnects. 62

13 Minimum size (X1) std. cell average delay (in ps), assuming worst loading, at different corners. 63

14 Benchmarks used for evaluation evaluation. 67

15 Basic floorplan comparisons assuming both tiers have same performance. . 68

16 Basic floorplan comparisons for different degraded 3D options. The numbers are normalized to the respective 2D numbers in Table 15. 70

17 Impact of performance difference aware floorplanning (PDAFP). '-' indicates that point is not achievable within $\pm 10\% \ V_{DD}$. 72

18 Iso-power performance and iso-performance power results for all implementation flavors. 72

19 Placement results for the 128×4 multiplier block. 74

20 The various benchmarks considered in this section. 94

21 The impact of partition bin size on solution quality. 96

22 The impact of router-based MIV insertion. Entries marked with a * are unroutable. . . . 98

23 The impact of routability-driven partitioning on monolithic 3D IC designs. . 99

24 The impact of routability-driven partitioning for face-to-face designs. . . . 103

25 Overall Comparisons . . . 103

26 Comparison between 3D-Craft and Our Placer . . . 105

27 Comparison of single vs. multiple MIV/F2F insertion. Power values are reported in mW, and wirelength in meter. . . . 114

28 Comparison of two different types of 3D CTS. Power values are reported in mW, and wirelength in meter. . . . 114

29 Overall comparisons between 2D and different 3D implementation styles. Power numbers are in mW. . . . 116

30 Dual-Vt comparisons between 2D and different 3D implementation styles. Power is in mW. . . . 117

31 Material properties used in a mobile package. . . . 123

32 Benchmarks used. . . . 124

33 Design statistics of baseline 2D and 3D designs. . . . 126

34 The impact of IR-drop-aware partitioning. The PDN utilization is kept the same as the baseline designs. . . . 128

35 The impact of PDN optimization such that the IR-drop falls within the $45mV$ target. . . . 130

LIST OF FIGURES

1 The fabrication process of monolithic 3D ICs [2]. (a) The bottom tier is created the same way as 2D ICs. (b,c,d) Attachment of thin layer of silicon to the top of the bottom tier. (e) FEOL of top tier and creation of MIVs and top-tier contacts, and (f) BEOL processing of top-tier. 3

2 Various design styles available for monolithic 3D ICs. 4

3 Scan chain grown from (a) one direction, and (b) two directions. 9

4 Re-use of existing signal TSVs for scan chain 10

5 (a) A 3D scan chain, and (b) multiple fragments connected together 11

6 The impact of scan configuration on wirelength 12

7 The Structure of a 3D Integrated Circuit 13

8 An IEEE 1500 Wrapper Boundary Register capable of launching a transition on CFO. The abbreviations used are S-shiftWR, C-captureWR, T-transferWR, U-updateWR 14

9 The DfT Architecture for Transition Delay Fault Testing of 3D ICs, showing only the data path and serial operation 15

10 (a) A 0 to 1 Transition launched from WBR on Top Die (no TSV testing), (b) An equivalent 0 to 1 Transition launched from WBR on Bottom Die (with TSV testing) 16

11 (a) Post-bond test of bottom die, (b) Post-bond test of top die with TSV test. Solid red lines indicate flow of scanned data, and dashed blue lines indicate flow of data to and from WBRs in the launch-capture window . . . 17

12 Damage caused to the probe pad after a single probe touchdown [27]. . . . 18

13 Layout images of (a) probe pads and TSVs, (b) P/G TSVs and P/G wire detours, (c) signal TSVs and P/G wires. P/G wires can be routed over signal TSVs. 19

14 (a) Candidate locations for probe pads, (b) Sample horizontal and vertical power/ground pads, as well as signal pads, (c) 4 power probe pads placed in a 2 $\times$2 horizontal configuration, and (d) in a vertical configuration 19

15 The overall design Flow. Yellow indicates inputs to the flow, green boxes are custom scripts, blue indicates use of Synopsys tools, and red the use of Cadence tools 20

16 A sample waveform obtained during testing, designed with four scan chains 22

17 GDSII images. (a) A close up of a TSV and its WBR, (b) IEEE 1500 Instruction Register Chain, (c) zoom out shot of the top metal layer of the top die, showing TSV landing pads and probe pads 22

18 Various overheads involved in adding wrappers for (a) FFT and (b) Jpeg . . 23

19 Total Power comparison among (1) pre-bond, (2) post-bond without TSV test, and (3) post-bond with TSV test under five different test vectors. . . . 25

20 Pre-bond IR-drop under different probe pad configurations and test vectors for FFT (a, b, c) and Jpeg (d, e) 26

21 IR-drop maps before (= a, c) and after (= b, d) probe pad optimization. . . . 27

22 Comparison between pre-bond and post-bond IR-drop. (a) FFT, bottom die, (b) FFT, top die, (c) Jpeg bottom die, (d) Jpeg, top die 28

23 (a) GDSII screen shot of a single die of a block-level 3D IC (b) Zoom in shot of the boxed TSV block in (a) 29

24 Three different circuits considered for die-level partitioning of a two-die stack. (a) A homogeneous stack, (b & c) Two different partitions of a heterogeneous stack. A larger number implies the die is more complex. . . 31

25 Circuits considered for die-level partitioning of multi-die stacks. (a - c) three die stack, (d - f) four die stack. A larger number implies the die is more complex. 32

26 The variation in test time observed for a two-die stack starting with ckt2_p2 and performing 1000 different random moves. 50 test-pins and 2 different test-TSV constraints are assumed. 35

27 Comparison between the measured test time and approximate lower bound of test time (= Equation 13) for a 2 die stack. The number of test pins is 50. 39

28 Variation in test time observed while performing 1000 random moves, starting with ckt3_p1. The test time is computed assuming 50 test-pins, and 2 different uniform TSV constraints (20 vs 50 per-die). 40

29 Comparison between the measured test time and approximate lower bound for a four-die stack. The test pin constraint is assumed to be 100. 43

30 Comparison of the variation in test time observed between moves involving the bottom die (= D1 moves), and all other moves. The numbers are reported for four-die implementations of (a,b) b19, (c,d) des_perf. 44

31 Comparison of theoretical and experimental threshold complexity factors under various TSV and pin constraints. (a,b) Two-die stack, (c,d) Four-die stack. 45

32 The variation in $TSV_{t,po}$ observed while performing 1000 different random moves, assuming 50 test-pins. (a) b19 two-dies, (b) b19 four-dies, (c) des_perf two-dies and (d) des_perf four-dies. 46

33 The design flow to obtain a 3D floorplan, assuming hard blocks. 49

34 Histogram of the longest path delay through inter-block nets of a benchmark. 51

35 Iterative MIV planning algorithm for soft blocks. 54

36 Illustration of MIV planning for soft blocks. (a) Initial estimated MIV locations (b) After one iteration of MIV planning. 54

37 Our design flow used to get post-layout simulation results. 56

38 Sample layouts for cf_rca_16 testcase, along with select block designs, and zoomed in shots of TSVs and MIVs 57

39 Copper vs. Tungsten resistivity at different wire widths. 61

40 IV curves of nominal and degraded transistors. 62

41 Synthesis results of “des3” benchmark for different degradations. 64

42 The proposed inter-tier performance difference aware floorplanner. 65

43 Floorplan screenshots of “des3” when the top tier is at the TTm20p corner. (a) Without performance difference aware floorplanning, and (b) With performance difference aware floorplanning. 66

44 Power-performance trade-off curves assuming that both the tiers have identical transistors and interconnects. 69

45 Power-performance trade-off curves assuming degraded transistors and interconnects. Dashed lines represent non performance difference aware floorplanning and solid lines represent performance difference aware floorplanning. 71

46 3D placement layout snapshots of one 128×4 multiplier block within the “mul128” benchmark. 74

47 Power-performance trade-off curves for the 128×4 multiplier block. 75

48 The design flow used for gate-level M3D placement. 77

49 Placement-aware partitioning. A modified 2D engine is used to place all the gates into half the area, and then partitioned with area balance in each bin. 79

50 Handling pre-placed memory macros (a) Initial pre-placed locations, (b) Projection of both tiers onto the same plane, and (c) Modifying the target density to represent memory locations. t'_d is the target density in the modified 2D placement and t_d is the required target density in the final M3D design. 81

51 Construction of a 3D RST. (a) The points to be routed. (b) Project to 2D and construct a 2D RSMT. (c) Expand the 2D RSMT to a 3D RST. (d) If a cell changes tier, the 2D RSMT can be re-used. 84

52 A legal route from A to B in a $4 \times 3 \times 2$ grid. The top-view is limited to two bends, while the unfurled view can have unlimited bends. 85

53 A view of the top metal layer that contains MIV landing pads. (a) A 2D wire on the top metal layer blocks potential MIV landing pad slots. (b) If MIVs connect to cells outside the current bin (external), they block other MIVs. If MIVs connect to cells within the current bin (internal), they do not block other potential MIV slots. 87

54 An overview of the router-based MIV insertion methodology. (a) The technology and macro LEF are modified to represent a two-tier monolithic 3D IC. (b) The structure that is fed into the commercial router, which is then routed. The MIV locations are extracted and separate verilog/DEF files are created for each tier. 92

55 Screenshots of router-based MIV insertion (a) All the gates are placed in the same placement layer, but no overlap exists in the routing layers. (b) The result after routing. The MIV locations are highlighted in red. . . . 93

56 Manual partitioning of the memories in the OS_T2 benchmark. The memories belonging to each sub-module are partitioned, and placed in a configuration similar to that in 2D. 95

57 Supply, demand, and overflow maps of the mul_64 benchmark for min-cut based partitioning solution. If interdependent supply/demand is considered, a significant reduction in supply in densely wired areas is observed, leading to more overflow. 99

58 The impact of reducing the metal layer count. “Tm1” (“Bm1”) stands for one metal layer removed from the top (bottom) tier. 101

59 (a) Monolithic 3D integration, and (b) Face-to-face 3D integration. MIVs are limited to whitespace, while F2F vias are not. 102

60 Comparison of 2D, partition-then-place, and placement-aware partitioning methods. 106

61 The overall CAD methodology flow used in this paper. 107

62 Isolating the memory pins by shrinking the memory footprint. (a) Initial memory footprint, and (b) Memory footprint reduced to size of filler cell. . 109

63 Pre-placed memory is flattened to get a shrunk 2D footprint, on which 2D P&R is performed. This is then partitioned to get a monolithic 3D solution. 110

64 Two different types of 3D CTS possible (a) One clock tree per tier for each gating group (source-level), and (b) The entire backbone is fixed onto tier 0 (leaf-level). 111

65 The proposed CTS methodology (a) The clock backbone in tier 0, and (b) Zoom-in shot of leaf-level flip-flops in both tiers connected to a leaf clock buffer in tier 0. 112

66 Two types of MIV insertion for a 3D net (a) Single, (b) Multiple 113

67 Resistive equivalent circuits for IR-drop and thermal in a conventional and mobile package. Moving high power cells to the tier close to package helps alleviate IR-drop. In a mobile package, the temperature increase is much smaller than in a conventional package. Resistance is in $m\Omega$, and thermal resistance in $^{\circ}C/W$. 118

68 The design flow used for IR-drop-aware partitioning. 120

69 (a) A PDN structure in monolithic 3D. Red wires represent VDD and blue wires represent VSS, (b) The power mesh showing the top and intermediate metal layers, (c) Zoom-in shot of PDN MIV arrays showing only the intermediate mesh layer and local cell rails. 122

70 A structure of a mobile package in 3D VLSI [1]. 123

71 Sensitivity of tier IR-drop to change in tier power for (a) crossbar, and (b) jpeg. 126

72 IR-drop maps for crossbar benchmark. (a) baseline, (b) our IR-drop-aware partition, where tier 0 has 60% of the chip power. 129

73 The impact of PDN optimization on the crossbar benchmark. IR-drop aware partitioning is able to achieve the same IR-drop target as the baseline partition while using significantly fewer PDN resources. 131

74 The impact of changing the target power of the bottom tier on the temperature of the crossbar benchmark. Even if the bottom tier has 70% of the chip power, the temperature increase is $< 1^{\circ}C$. 132

SUMMARY

The objective of this research is to develop physical design methodologies for monolithic 3D ICs and use them to evaluate the improvements in the power-performance envelope offered over 2D ICs. In addition, design-for-test (DfT) techniques essential for the adoption of shorter term through-silicon-via (TSV) based 3D ICs are explored.

Testing of TSV-based 3D ICs is one of the last challenges facing their commercialization. First, a pre-bond testable 3D scan chain construction technique is developed. Next, a transition-delay-fault test architecture is presented, along with a study on how to mitigate IR-drop. Finally, to facilitate partitioning, a quick and accurate framework for test-TSV estimation is developed.

Block-level monolithic 3D ICs will be the first to emerge, as significant IP can be reused. However, no physical design flows exist, and hence a monolithic 3D floorplanning framework is developed. Next, inter-tier performance differences that arise due to the not yet mature fabrication process are investigated and modeled. Finally, an inter-tier performance-difference aware floorplanner is presented, and it is demonstrated that high quality 3D floorplans are achievable even under these inter-tier differences.

Monolithic 3D offers sufficient integration density to place individual gates in three dimensions and connect them together. However, no tools or techniques exist that can take advantage of the high integration density offered. Therefore, a gate-level framework that leverages existing 2D ICs tools is presented. This framework also provides congestion modeling and produces results that minimize routing congestion. Next, this framework is extended to commercial 2D IC tools, so that steps such as timing optimization and clock tree synthesis can be applied. Finally, a voltage-drop-aware partitioning technique is presented that can alleviate IR-drop issues, without any impact on the performance or maximum operating temperature of the chip.

CHAPTER I

INTRODUCTION

Technology scaling has been the fundamental driver of the semiconductor industry over the last few decades. Each new technology generation delivers chips that are not only smaller and faster, but also cheaper. However, scaling brings with it an exponential increase in fabrication complexity. Devices today are no longer planar, and finFET structures have become mainstream. Today's extremely small geometries ideally require advancements in lithography such as extreme-ultraviolet lithography. However, delays in its deployment have led to the necessity of stop-gap solutions such as double and triple patterning. This not only increases mask and fabrication cost, but also increases design cycle time. All this additional complexity has led to speculation that cost no longer scales below $28nm$.

These issues have led the industry to rethink the direction of technology scaling. Typically, as chips shrink, the devices get smaller and faster, but the interconnects become more resistive and slower. In older nodes, the interconnect delay was such a small portion of the total delay that this could be neglected. Today, however, the interconnect delay is dominant. This has led to three dimensional integrated circuits (3D ICs) being proposed as a solution to the interconnect bottleneck. In 3D ICs, devices are placed on multiple layers, instead of just one, and connected together. This reduces the length of the on-chip interconnect, squeezing additional performance out of the same device generation.

One of the first techniques developed to enable 3D ICs was through-silicon-vias (TSVs). Two or more layers of devices are fabricated, TSVs created on the dies, and then each die is aligned and bonded. This technology is relatively close to market, and design-for-test (DfT) is one of the last challenges facing its adoption. However, the quality of TSV-based 3D ICs

strongly depends on the TSV dimensions and parasitics, and they do not solve all interconnect issues. Their relatively large pitch and parasitics limit then to memory-on-logic or large logic-on-logic designs with relatively small number of global interconnects [66, 15].

An emerging alternative is monolithic 3D integration (M3D), where the tiers are fabricated sequentially, one on top of another, and connected together using monolithic inter-tier vias (MIVs). Since no die alignment is required, these MIVs are roughly the same size as local vias. Overall, monolithic 3D ICs offer several advantages over TSV-based 3D ICs: (1) the small size of MIVs enables ultra-high integration density, considerably reducing silicon area and cost, (2) the significantly reduced MIV parasitics help improve the power-performance envelope, and (3) the manufacturing process is entirely foundry-driven, and does not involve a packaging house for the processing of backside redistribution layers and micro-bumps. This enables tighter process control, potentially leading to a faster ramp-up once the technology is mature.

This section now presents an overview of the fabrication techniques and design styles available for monolithic 3D ICs, and then outlines the contributions and the structure of the rest of this dissertation.

1.1 Overview of Monolithic 3D ICs

1.1.1 Fabrication Techniques

The first technique developed to fabricate monolithic 3D ICs was to fabricate the bottom tier as usual, and then to deposit a thin-film of amorphous silicon on top of it. Existing know-how was then used to fabricate thin-film-transistors (TFT) on the top tier [25, 48]. However, the problem with this technique is that amorphous silicon leads to severely degraded transistors. Next, attempts were made to crystallize the amorphous silicon on the top tier using lasers [26, 18]. This, however, leads to islands of crystalline silicon with unpredictable device behaviour at these island boundaries. Batude *et al.* were the first to propose a process that produces extremely high quality crystalline silicon on the top tier,

which allows the fabrication of general logic [3, 2]. The rest of the dissertation assumes this process, and an illustration of it is given in Figure 1.

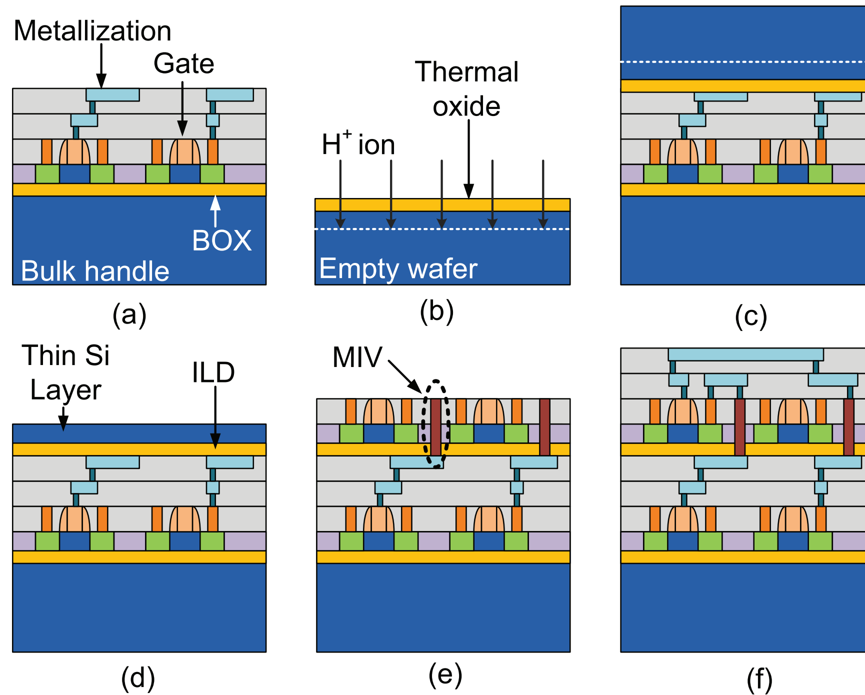


Figure 1: The fabrication process of monolithic 3D ICs [2]. (a) The bottom tier is created the same way as 2D ICs. (b,c,d) Attachment of thin layer of silicon to the top of the bottom tier. (e) FEOL of top tier and creation of MIVs and top-tier contacts, and (f) BEOL processing of top-tier.

This is a silicon-on-insulator (SOI) process, and the bottom tier is fabricated similar to a 2D IC (Figure 1(a)). Next, a thermal oxide is grown on an empty wafer, and H^+ ions are implanted just below the silicon surface at a constant depth (Figure 1(b)). The thickness of the oxide determines the buried oxide (BOX) thickness, and the depth of ion-implantation determines the active silicon thickness. This new wafer is then flipped and bonded to the top of the bottom tier using a low temperature oxide bonding process (Figure 1(c)). The excess silicon is then sheared of at the implant line, and polished using chemical-mechanical-polishing (CMP) to give an extremely high quality single-crystal silicon layer. The gates are formed on the top tier, and the MIVs are created with the contact mask of the top tier (Figure 1(e)). Finally, the metallization of the top tier is created (Figure 1(f)).

1.1.2 Design Styles

Monolithic 3D ICs were first applied to SRAM and FPGA design, where the masks are extremely regular, and full-custom design techniques are easily applied. Jung *et al.* demonstrated a 3D SRAM fabricated using a TFT layer on the top tier [25]. Naito *et al.* presented a monolithic 3D FPGA design using a TFT configuration SRAM over bulk CMOS logic [48]. Jung *et al.* also demonstrated a high-performance cost-effective DDR3 SRAM using epitaxial growth [26]. This technology also allows heterogeneous integration, such as that demonstrated by Golshani *et al.* , where a photodiode array was stacked onto SRAM for image sensing applications [18]. With respect to design, Liu and Lim evaluated several design options for 3D SRAM including separating the PMOS and NMOS into different tiers, and changing transistor and metal layer counts [40]. However, none of these works considered general logic, where physical design techniques becomes essential. In general, monolithic 3D ICs can be divided into three design styles as shown in Figure 2.

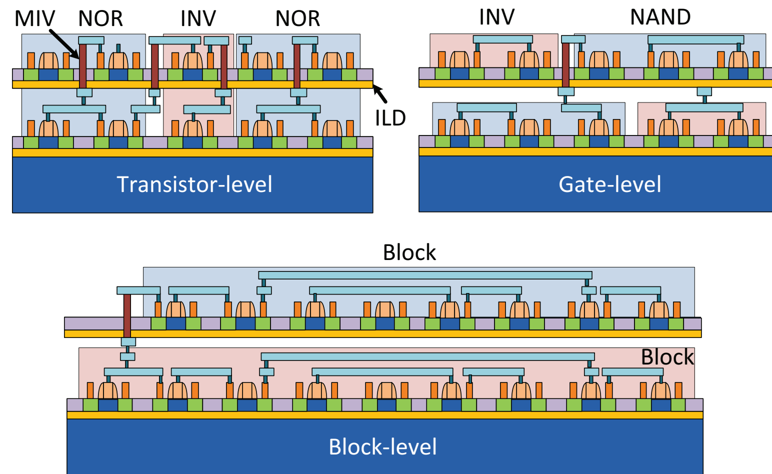


Figure 2: Various design styles available for monolithic 3D ICs.

Transistor-level integration is the most fine-grained technique [4, 33, 39, 34], where the PMOS and NMOS within standard cells are placed on different tiers. It has the advantage that the PMOS and NMOS fabrication process can be optimized separately. However, this style requires redesign and re-characterization of the standard cells themselves, which

takes significant effort. In addition, the standard cell footprint does not reduce by 50% in 3D due to the mismatch in the PMOS and NMOS sizes, as well as because MIVs are required within the cell itself. Lee, Morrow, and Lim have demonstrated that one of the main advantages of this design style is that the redesigned standard cells can be directly fed into existing 2D tools [34].

Since re-designing existing logic, memory and IP blocks for 3D incurs significant design overhead and cost, near-term 3D ICs will focus on reusing existing 2D blocks. In block-level monolithic 3D ICs, functional blocks are floorplanned onto different tiers. This style has the benefit of IP reuse, but does not fully take advantage of the fine-grained nature of MIVs. There has been no prior work in designing block-level monolithic 3D ICs.

The last design style is gate-level monolithic 3D ICs, where existing standard cells and memory can be placed on multiple tiers, and connected together using MIVs. The advantage of this style is that it offers the reuse of existing cells, zero total silicon area overhead (unlike transistor-level), and a sufficiently high integration density to obtain significant power benefits (unlike block-level). The only prior work in this design style is [4], where the authors provide a rudimentary design flow that is not capable of handling any hard macros such as memory, and therefore cannot be applied to real designs.

Therefore, for general logic designs, physical design for transistor-level monolithic 3D ICs have been explored, while there is a complete lack of CAD tools and methodologies to design real world block-level and gate-level monolithic 3D ICs.

1.2 Organization and Contributions

This research first explores design-for-test (DfT) techniques crucial to the commercialization of short term TSV-based 3D ICs. Next, it presents a complete sign-off physical design framework to take an RTL description of a circuit, and implement it in either a block-level or gate-level monolithic 3D IC. Each of these is organized into a self-contained chapter, and the contributions of this dissertation are as follows:

Design-for-Test for TSV-based 3D ICs is presented in Chapter 2. This chapter first presents a technique to construct 3D scan chains, that unlike previous works, is pre-bond testable. Next, as 3D ICs need to be tested at the rated frequency, this work presents the first transition-delay-fault capable test architecture for 3D ICs. In addition, since IR-drop is an issue during transition testing, techniques to mitigate IR-drop are presented. Finally, this chapter presents techniques to quickly and accurately estimate the test time of a given 3D IC partition. This estimate can be used during the partitioning process to assess the total number of test TSVs required by the partition under consideration.

Physical Design for Block-level Monolithic 3D ICs is discussed in Chapter 3. First, a floorplanning framework is presented, and it is demonstrated that this engine produces results comparable to commercial 2D engines. Inter-tier performance differences that arise due to an immature fabrication process is discussed, and two options to mitigate these differences are discussed and modeled. A performance-difference aware floorplanner that uses these models to produce high quality monolithic 3D floorplans is also presented.

Physical Design for Gate-level Monolithic 3D ICs is covered in Chapter 4. This chapter first presents a technique to modify existing academic 2D engines, and couple them with a placement-aware partitioning step to obtain high-quality monolithic 3D IC placement solutions. It also discusses a technique to use commercial routers for MIV insertion. In addition, it presents a technique to utilize commercial 2D engines instead of academic ones. Finally, an IR-drop-aware partitioner that reduces the power and IR-drop of a monolithic 3D IC without increasing the maximum operating temperature of the chip is developed.

Conclusions and Future Directions are discussed in Chapter 5. This chapter summarises all the work presented in this dissertation and goes over future research directions that will help in designing better quality industrial-sized monolithic 3D systems-on-a-chip.

CHAPTER II

DESIGN-FOR-TEST FOR TSV-BASED 3D ICS

TSV-based 3D ICs are manufactured by fabricating each die separately, thinning the dies containing TSVs, and stacking them all together. Due to the additional manufacturing steps of thinning and stacking, additional defects could be introduced into the circuit. Therefore, these 3D ICs need to be tested both before stacking (pre-bond), and after stacking (post-bond). Testing of TSV-based 3D ICs is one of the last EDA challenges facing their widespread adoption [62], and some of the challenges facing 3D test were enumerated in [32].

Wu *et al.* [63] compare several scan-chain schemes, and provide genetic and ILP based algorithms for post-bond test. Zhao *et al.* [67] provide a scheme for clock tree synthesis to facilitate pre-bond test. At the architectural level, Lewis and Lee [35] proposed a scan island based methodology to test incomplete circuits during pre-bond test. This architecture is similar to IEEE 1500, and a pre-bond testable architecture based on extensions to IEEE 1500 was formalized in [44, 46, 47]. The authors of [22, 23] provide test architecture design for 3D SoCs, and Lee *et al.* provide an architecture that supports different test-access-mechanism (TAM) widths for pre-bond and post-bond test [36].

In this chapter, three different aspects of DfT for TSV-based 3D ICs are presented. First, a pre and post-bond testable scan chain design scheme is discussed. Next, a transition-delay-fault capable test architecture that can test 3D ICs at the rated functional frequency is presented. Since voltage-drop (IR-drop) becomes an issue at the functional frequency, this chapter also discusses power-delivery issues and IR-drop mitigation during test. Finally, a theoretical framework to quickly estimate the test time of a given 3D IC partition is included. Typical use cases and benefits of such a framework are also demonstrated.

2.1 *Scan-Chain Design for 3D ICs*

Constructing a 3D scan chain (i.e, goes across tiers) has several advantages over constructing one 2D scan chain per tier and stitching them together. However, since a 3D scan chain relies on the use of TSVs, and since TSVs occupy significant silicon area, the number of scan TSVs that can be used is limited. Wu *et al.* [63] have demonstrated that 3D scan chains give up to a 40% reduction in the scan wirelength. This can significantly improve the speed of the scan chain, and reduce the test time of the circuit. However, the approach presented in their work does not support pre-bond test, and assumes that the dies will be tested only after bonding. This project demonstrates a scan-chain construction approach that makes use of 3D scan chains, and is also pre-bond testable.

2.1.1 3D Scan Chain Construction

This section presents a greedy heuristic to construct a 3D scan chain while minimizing its wirelength. The input constraints are the maximum number of scan TSVs that can be used, the location of all the flip-flops, and a fixed scan-in and scan-out pin. The heuristic is presented in Algorithm 1.

Algorithm 1: Greedy algorithm to construct a 3D scan chain

1. $C \leftarrow \{c_1, c_2, \ldots, c_{k-1}\}$;
2. $X \leftarrow \{x_0, x_1, x_2, \ldots, x_m, x_{m+1}\}$;
3. $\forall i, j$ `Initialize` (`Cost` (i,j)) ;
4. $M = \{x_0, x_{m+1}\}$;
5. $u \leftarrow x_0$ $v \leftarrow x_{m+1}$;
6. **while** $M \cap X \neq X$ **do**
7. $u' =$ `Min` (`Cost` (u,j)) , $j \notin M$;
8. $M = M \cup j$;
9. $u = u'$;
10. $\forall i, j$ `Update` (`Cost` (i,j)) ;
11. $v' =$ `Min` (`Cost` (v,k)) , $j \notin M$;
12. $M = M \cup k$;
13. $v = v'$;
14. $\forall i, j$ `Update` (`Cost` (i,j)) ;
15. **end**

Here, C represents the TSV constraint for each die, and there are k dies. Assuming face-to-back (F2B) bonding, TSVs are absent on the last die, and there are $k-1$ constraints. X represents the set of all scan cells, which has size m. x_0 represents the scan-in pin, and x_{m+1} represents the scan out pin. Next, the cost function between two cells is initialized. This cost function is given by Equation (1), where z represents all dies between x_i and x_j, and R_z represents the remaining number of TSVs that can be used in that die without violating the TSV constraint.

$$Cost(i,j) = \begin{cases} d_{ij} & \text{i,j in same die} \\ \dfrac{d_{ij}}{\min R_z/C_z} & \text{otherwise} \end{cases} \tag{1}$$

Set M represents the set of marked cells, and the scan-in and scan-out pin are initially marked. Next, the scan chain is grown from two sides, both from the scan-in and the scan-out pins. Each iteration picks the cell with minimum cost, and this process continues until all cells are marked. The cost function is dynamically updated, and TSVs become more expensive as the TSV constraint is approached. Eventually, the cost of using a TSV becomes infinity once the TSV constraint is reached. It is important to note that when this happens, it may not be possible to stitch all the scan cells without using more TSVs due to the presence of isolated chains. In this case, extra TSVs may be used, which is guaranteed to not exceed two TSVs per die, and the constraints can be adjusted appropriately. Although it is possible to grow the scan chain from one direction only, growing it from two directions usually results in smaller scan wirelength, as as shown in Figure 3.

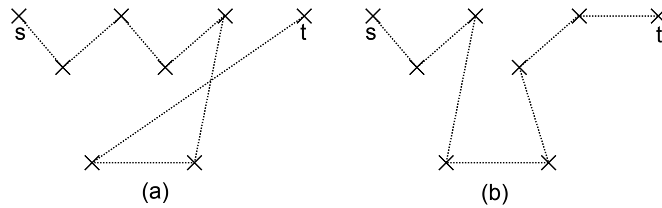


Figure 3: Scan chain grown from (a) one direction, and (b) two directions.

2.1.2 Reuse of Signal TSVs

So far, the assumption has been that a dedicated scan TSV is required when a scan chain goes from one die to another. In a scan chain, the output of a flip-flop is connected to the scan input of the next flip flop, as well as to some combinational logic that is of no consequence during the test mode. It might be possible that a flip flop drives some combinational logic on another die through an existing signal TSV. In such a case, an additional dedicated scan TSV is not required, and the existing signal TSV can be reused. A careful choice of scan ordering can make use of several existing signal TSVs, thereby reducing the overall scan chain wirelength, without suffering the penalty of inserting a large TSV into the layout. An example of signal-TSV reuse is shown in Figure 4.

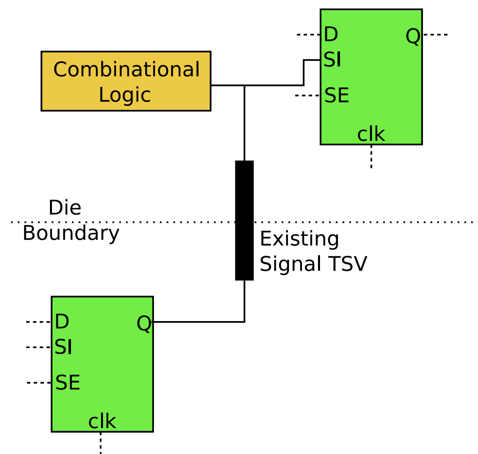


Figure 4: Re-use of existing signal TSVs for scan chain

2.1.3 Broken Scan Chains

Once a 3D scan chain is inserted into the design, it is used during post-bond test, and its scan-in and scan-out pins are accessed through solder bumps. However, if pre-bond test is to be performed, the scan chains on each die are broken into a number of fragments, and cannot be used as-is. It is not feasible to probe all these fragments as probe needles are usually large and their number is quite small. Thus, it becomes necessary to stitch together different fragments as shown in Figure 5 so that the pre-bond test-pin count is reduced.

This can be achieved using tri-state buffers to stitch together the broken fragments, and enabling them using a pre-bond test signal.

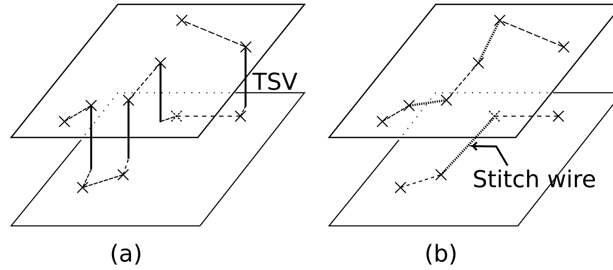


Figure 5: (a) A 3D scan chain, and (b) multiple fragments connected together

2.1.4 Experimental Results

Initially scan cells are inserted into the 2D netlist, either during or after synthesis. Next, the original netlist is partitioned into as many dies as required, and individual netlists are obtained for each die. Each die is then placed individually using Cadence Encounter to get initial rough locations of scan flip-flops. Scan chains are then stitched together using the greedy algorithm discussed earlier. This process introduces additional scan TSVs into the design, and placement is again carried out to accommodate them.

The greedy heuristic for scan chain insertion was implemented in C++, and a FFT circuit from [51] is chosen for analysis. Synthesis was carried out in Synopsys Design Compiler using NCSU $45nm$ technology. The design was placed in two dies, and the number of signal TSVs is chosen to occupy around 20% of the entire die area. Statistics about the design used are as follows: the number of gates is $400,213$, signal TSVs is 2953, and flip-flops is $75,723$. The TSV is assumed to have a diameter of $6\mu m$, with a height of $50\mu m$. The inserted wrapper scan elements occupy 1.96% of the total die area, and have a total stitched wirelength of $75054\mu m$.

In order to study the impact of the number of scan TSVs on the wirelength of the design, three different scan chains are constructed, as shown in Table 1. Since a 3D scan chain cannot be constructed without any TSVs, the "scan0" case has two test TSVs inserted,

which is the minimum number required. Column 3 shows that even without using any specific algorithm to re-use existing signal TSVs, it is possible to re-use around 2% of the TSVs required for the scan chain. The number of scan-chain fragments formed per die is exactly half of the number of TSVs, and Column 4 gives the amount of additional wirelength that is required to stitch all of them together into a single chain. With an increase in the number of fragments, the wirelength required to stitch them together also increases.

Table 1: Statistics for different scan chain configurations.

Name	No. TSVs	#TSV reused	Stitch WL (μm)
scan0	2	0	4.75
scan100	100	2	26595
scan200	200	4	34296

The impact of the scan chain TSV count on the scan wirelength and the total wirelength of the 3D design is plotted in Figure 6(a). First, it is observed that an increase in the number of scan TSVs always helps reduce the scan wirelength. However, adding more scan TSVs does not always reduce the signal and total wirelength. Beyond a certain point they start to worsen. The initial improvement is achieved because the lower scan wirelength reduces the routing congestion. With a further increase in the number of TSVs, either the die area or standard cell density increases. If the die area increases, the average distance between gates increases, increasing the overall wirelength. An increase in the cell density increases routing congestion, and hence wirelength.

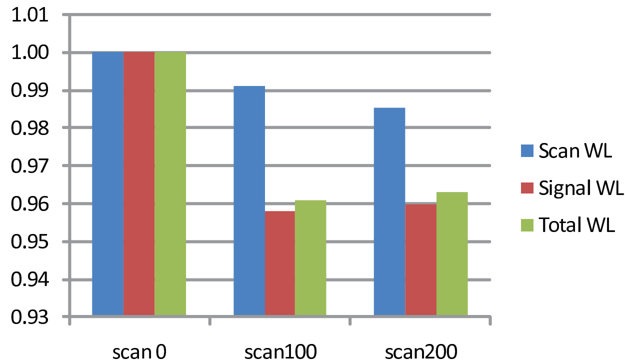


Figure 6: The impact of scan configuration on wirelength

2.2 *Transistion-delay-fault Testing for 3D ICs with IR-drop Study*

One of the reasons 3D ICs are being explored is because they are expected to be faster than 2D ICs. Therefore, it becomes essential to test them at the rated functional frequency, and make sure that they work. While there exists literature that supports transition delay fault testing of 2D SoCs [41, 7], no prior work has looked at transition delay fault testing for 3D ICs. This section first presents a DfT architecture that supports transition delay fault testing of 3D ICs. It supports both pre-bond and post-bond transition testing. In addition, it supports transition testing of TSVs after bonding.

In a 3D IC, only one die has C4 bumps, and all other dies have no direct test access. During pre-bond test of these dies, no wire bond pads exist, and it becomes necessary to add large probe pads into the layout to facilitate probe needle touchdown, as shown in Figure 7. This section discusses how these probe pads can be added into the layout, and how they fit into the transition delay fault test architecture.

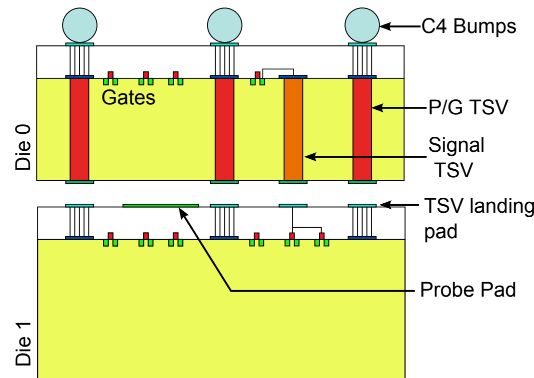


Figure 7: The Structure of a 3D Integrated Circuit

Finally, since transition test is carried out at the rated frequency of the chip, excessive voltage drop (IR-drop) may occur. This is because test pattern generation tools aim to test as many faults as possible with each pattern, leading to large portions of the chip switching at the same time. This section also discusses creating a power delivery network (PDN) that can support transition test, including the addition of power/ground probe pads, and techniques to mitigate IR-drop.

2.2.1 Transition-delay-fault Architecture

The application of a transition fault vector to a circuit requires two cycles. The first cycle triggers a transition (launch) at the location to be tested, and the second cycle (capture) captures the response to this transition. The IEEE 1500 Wrapper Boundary Registers (WBR) specified in [44], cannot directly be used as it only supports the application of a single bit to a primary input, while two bits are required to launch a transition. Instead, a three flip flop IEEE 1500 WBR specified in [41] is used. Such a register is shown in Figure 8. This figure also explains abbreviations that will be used in the remainder of this section. Each flip flop is sensitive to a different combination of IEEE 1500 control signals, which are indicated above the clock. To apply a transition test, one bit is scanned into each of the SC and ST flip-flops, and applied sequentially through the Update register.

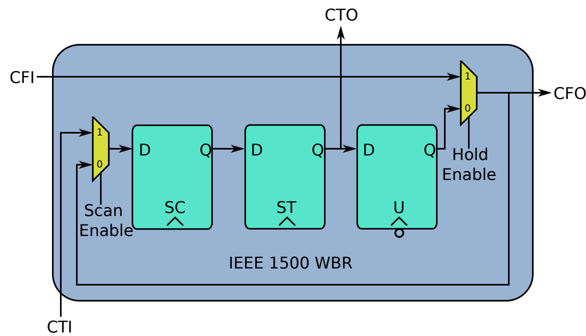


Figure 8: An IEEE 1500 Wrapper Boundary Register capable of launching a transition on CFO. The abbreviations used are S-shiftWR, C-captureWR, T-transferWR, U-updateWR

The overall transition fault DfT architecture is shown in Figure 9. This figure is simplified for illustration, and only the data path and a serial scan chain is shown. Parallel testing is essentially the same idea, but with a larger number of scan chains. Each TSV is equipped with a WBR, so that values can be scanned into it during test. Once the values are scanned in, the launch and capture clocks are applied, and the responses are scanned out. Each die is tested independently of the other, during both the pre-bond and post-bond tests. Each unwrapped die is equipped with an internal bypass, so that the internal scan chains can be bypassed, if desired. In order to transport data to and from the top die, the bottom

die is equipped with a multiplexer (elevator enable) to select the data from the top die. The various control signals are generated by the IEEE 1149.1 TAP controller.

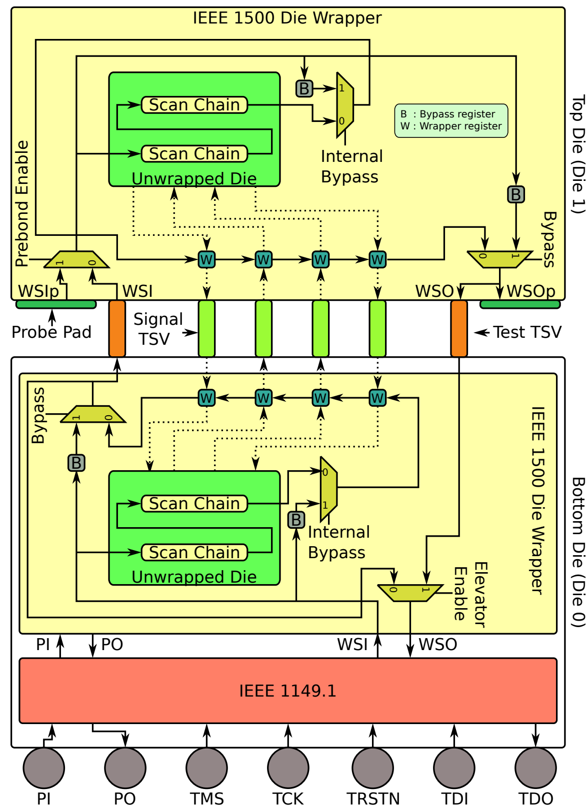


Figure 9: The DfT Architecture for Transition Delay Fault Testing of 3D ICs, showing only the data path and serial operation

This architecture is similar to that presented in [44], but with a few notable differences. The first one is that a transition fault capable WBR is used. The second is that this system has to support the transfer operation, in order to transfer data between the SC and ST registers. Therefore, an extra transfer signal is to be routed between the dies. However, the IEEE 1149.1 TAP controller does not natively support the application of delay tests, and two approaches to modify it exist in the literature. The first one uses the exit1-DR, exit2-DR and pause-DR states of the IEEE 1149.1 FSM to generate update, transfer and capture signals, while in delay test mode [41]. The second approach utilizes an additional TMS bit to change the state from update-DR to capture-DR within a single clock cycle [7]. The first approach is used, because additional package pins are undesirable.

TSVs also need to be tested at-speed in a 3D IC. For stuck-at fault testing, TSV testing is trivial. Each TSV has a WBR on either side, and TSVs can be tested by placing both dies in their respective extest modes. However, for transition testing, the time between the launch and capture pulses has to be of the order of the TSV delay. This is a few tens of picoseconds, and it is unreasonable to assume that the clock can be applied with such a high speed.

This section presents an alternate approach to test the TSVs after the dies have been bonded. Consider Figure 10(a). This represents the post-bond testing of the top die, with a transition launched from the WBR on the top die. Figure 10(b) shows the identical transition on the top die, but launched from the WBR on the bottom die. This transition would also occur on the TSV, and would hence test the TSV also. This implies that a test vector generated for the top die, but launched from the bottom die will also test TSVs. If, after bonding, the testing of the top die is performed exclusively through the WBRs of the bottom die, no additional patterns will be required, and all TSVs between the top and bottom die will be tested.

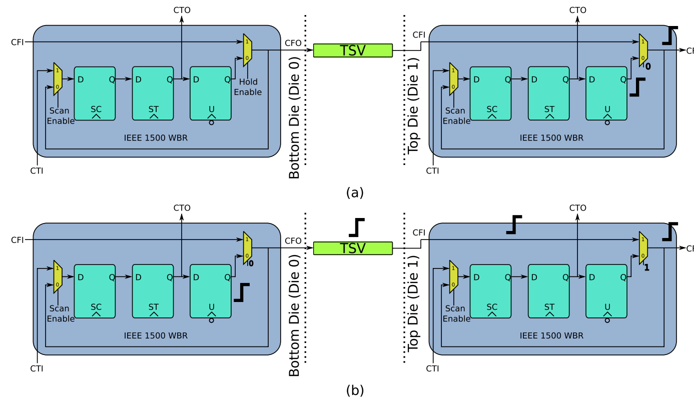


Figure 10: (a) A 0 to 1 Transition launched from WBR on Top Die (no TSV testing), (b) An equivalent 0 to 1 Transition launched from WBR on Bottom Die (with TSV testing)

In order to support TSV test, an additional mode of operation that configures the WBRs as shown in Figure 10 is required, which is called TSVtest. The default modes presented by [44] are serial/parallel , pre-bond/post-bond, intest/extest/bypass and turn/elevator. If

a die is placed into TSVtest, all WBRs facing the bottom die are made transparent. TSV testing can then be performed by placing the bottom die into extest, and the top die in the intest_TSVtest mode.

Two example modes of operation are shown in Figure 11. Figure 11(a) Shows the post-bond test of the bottom die. The instruction used is post-bond-intest-serial-turn. Figure 11(b) shows the post-bond testing of the top die with TSV test. Here the bottom die is programmed with post-bond-extest-serial-elevator, and the top die with post-bond-intest-serial-turn-TSVtest. The solid red lines show the flow of data scanned in, and the dashed blue lines show the data flow to and from the WBRs in the launch–capture window.

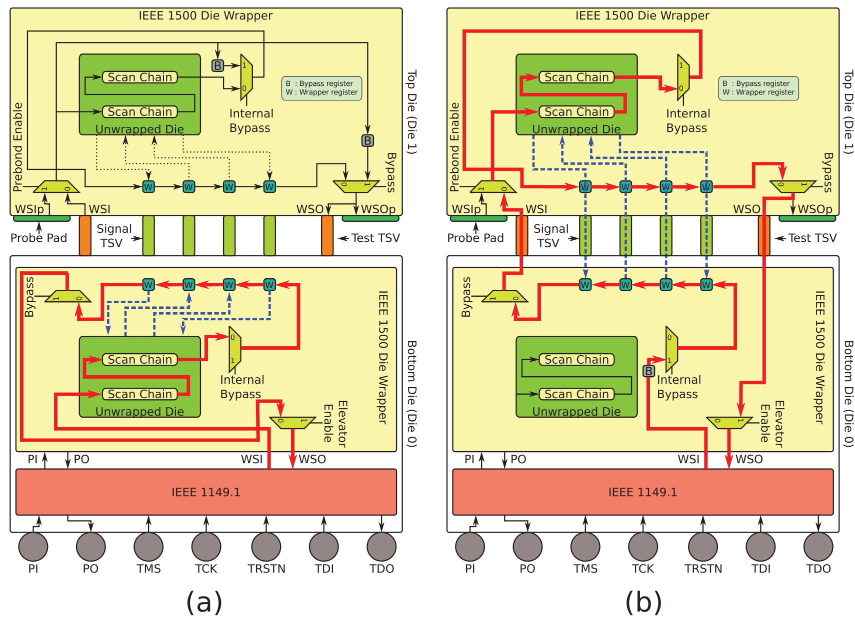


Figure 11: (a) Post-bond test of bottom die, (b) Post-bond test of top die with TSV test. Solid red lines indicate flow of scanned data, and dashed blue lines indicate flow of data to and from WBRs in the launch-capture window

2.2.2 Probe-pad Placement and PDN Design

Fine grained probe needles are unlikely to be available at least for another decade [53]. Today's probe pads are limited by available technology [43] to a minimum pitch of $35 - 40\mu m$ for cantilever probing, and $100\mu m$ for vertical probing with a minimum pad size of around $25\mu m$. As seen from Figure 7, not only do these probes occupy significant area

on the die in which it is placed, any TSVs in the previous die cannot be placed in the same location as the probe pad in order to avoid overlap with its landing pad. In addition, when the probe needle makes contact with the probe pad, it creates a scrub mark, which significantly affects its planarity, as shown in Figure 12 [27]. Therefore, several layout implications exist while adding probe pads, and their locations need to be chosen carefully.

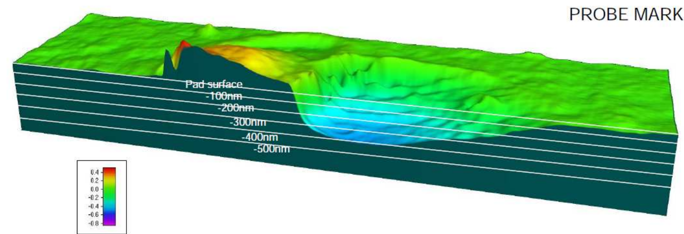


Figure 12: Damage caused to the probe pad after a single probe touchdown [27].

Probe pads can be divided into two categories – signal and power/ground (PG). Signal probe pads are needed as the top die requires test access during pre-bond test. Each IEEE 1500 data and control signal needs to be provided with its own probe pad. In addition to these, the die needs to be powered during test. Ideally, each PG TSV chosen for touchdown would have a PG probe pad directly on top of its landing pad. This would minimize the area overhead, as well as provide a low resistance connection for power delivery. However, the scrub mark will affect the TSV bonding process, and for the sake of reliability, a certain distance has to be maintained between the probe pad and the TSV landing pad. Figure 13 shows such an arrangement. This figure also shows how PG TSVs are created in the layout, and since they are quite large, how the thin PG rails detour around them.

This study focuses on circuits that have a regular power and ground TSV placement as shown in Figure 14(a). Since the power and ground TSVs form a regular array, the space in between them are candidate locations for probe pads. PG and signal probe pads can be placed in a subset of these candidate locations. An example is shown in Figure 14(b). Two choices exist when connecting a power probe pad to a power TSV – either a horizontal or a vertical configuration. This figure also shows two signal probe pads. To simplify the design

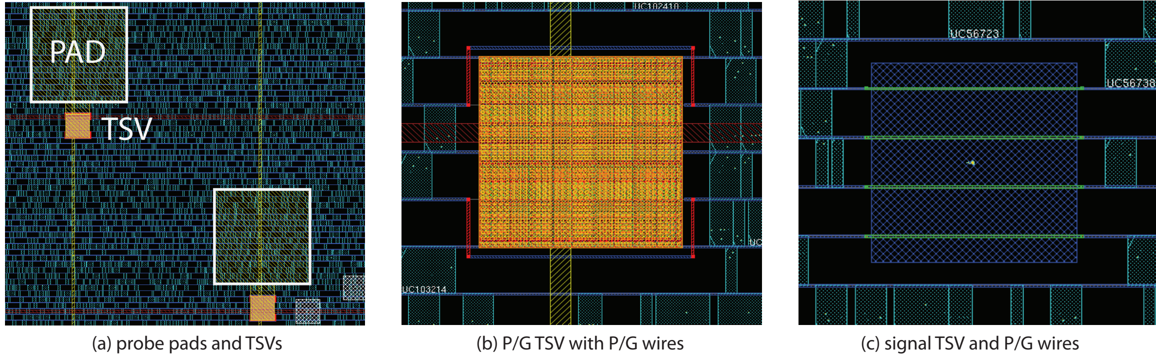


Figure 13: Layout images of (a) probe pads and TSVs, (b) P/G TSVs and P/G wire detours, (c) signal TSVs and P/G wires. P/G wires can be routed over signal TSVs.

process and reduce the search space, PG probe pads are placed in either the horizontal or the vertical configuration, but not both. Figure 14(c) shows how 4 power probe pads are placed in a 2×2 array in a horizontal configuration, and Figure 14(d) shows the same for a vertical configuration.

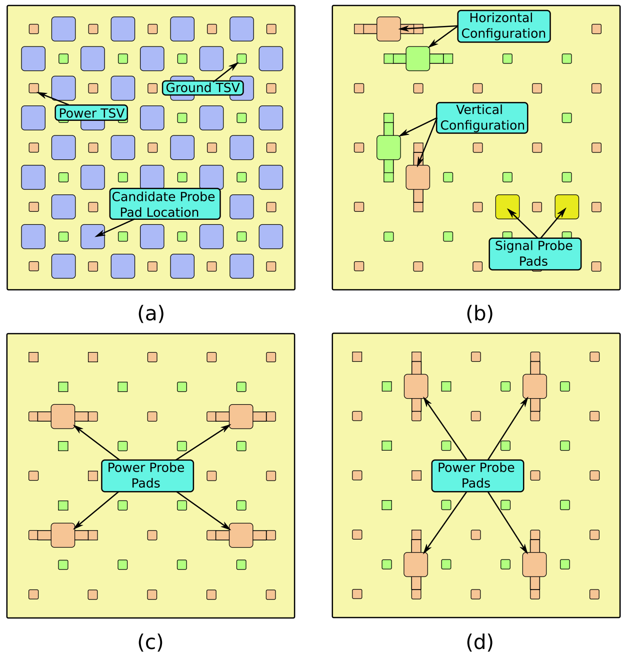


Figure 14: (a) Candidate locations for probe pads, (b) Sample horizontal and vertical power/ground pads, as well as signal pads, (c) 4 power probe pads placed in a 2×2 horizontal configuration, and (d) in a vertical configuration

2.2.3 Design and Analysis Flow

The design flow used in this section is shown in Figure 15. It can be broadly divided into two categories. The left column represents physical design, and the right column represents test related steps. Finally, IR-drop analysis is performed. Each step is explained individually below.

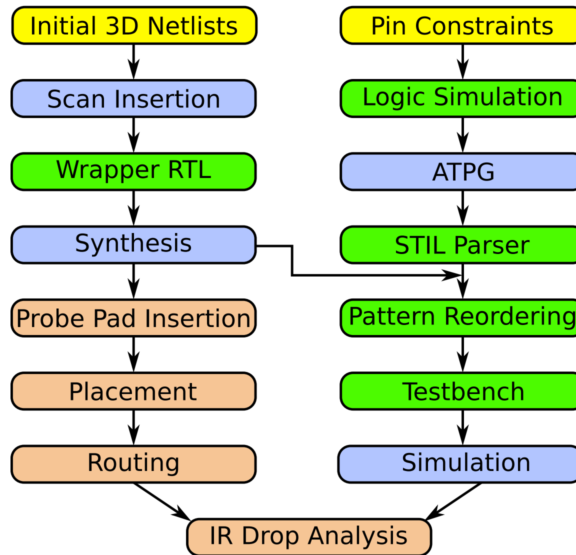


Figure 15: The overall design Flow. Yellow indicates inputs to the flow, green boxes are custom scripts, blue indicates use of Synopsys tools, and red the use of Cadence tools

With respect to physical design, the starting point is an initial 3D gate-level verilog netlist, generated by partitioning a 2D netlist. Synopsys Design Compiler is then used to insert as many scan chains per die as required. Custom scripts then take this netlist with scan chains, and generates the RTL for the IEEE 1500 wrapper. This is then re-synthesized using Synopsys Design Compiler. Probe pads are then inserted into the layout and treated as locations where other TSVs cannot be placed. The design is then placed and routed using Cadence Encounter.

The test related steps starts with pin constraints, which are any pins that need to be constrained to a certain logic value during the test mode (such as reset). Logic simulation is then performed on the bottom die to get the pin constraints on the top die. Using

this information, automatic test pattern generation (ATPG) is performed on both dies using Synopsys Tetramax. The output is STIL files, containing pattern information. These are parsed, and using the information about the wrapper chain ordering from the physical design stage, the bits in the test patterns are reordered . A testbench is generated, and simulated using Synopsys VCS. Using the routed result, and the VCD file generated from the testbench, IR-drop analysis is performed as described next.

For 2D IR-drop analysis, as is the case with all pre-bond testing, existing tools can simply be used. However, 3D IR-drop analysis is required to measure the post-bond voltage drop. Power simulations are first performed on a per-die basis using the switching activity from the VCD file, after annotating each die with TSV parasitics. The DEF files from both the dies are then combined into a single DEF file, treating the TSV as a via. This tricks the tool into believing that it is dealing with a 2D design, but with a higher number of metal layers. The power numbers generated earlier can then be used to perform 3D IR-drop analysis using Cadence Encounter.

2.2.4 Experimental Results

All required scripts were implemented in C++. The designs used are synthesized using the nangate $45nm$ technology library. The TSV diameter is assumed to be $4\mu m$, and its height to be $40\mu m$. The TSV landing pads size is assumed to be $7\mu m$, and the total TSV cell size including keep out zone is $8.4\mu m$. Power and ground TSVs are placed in a regular fashion, with a pitch of $130\mu m$. The TSV resistance, including contact resistance is considered to be $50m\Omega$. The probe pads are assumed to have a size of $40\mu m \times 40\mu m$, and that the minimum pitch is $100\mu m$.

Figure 16 shows a sample testing waveform of a design with four scan chains. During capture, the responses from the circuit are stored into the SC register, and the value of ST is don't care. Only the first vector scanned out exhibits this don't care, and all subsequent vectors have a junk value in the ST register. Sample layouts are shown in Figure 17.

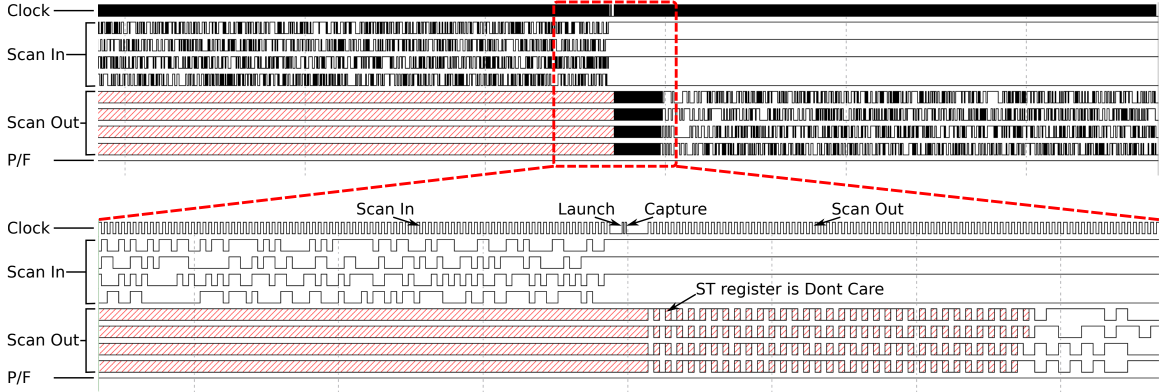


Figure 16: A sample waveform obtained during testing, designed with four scan chains

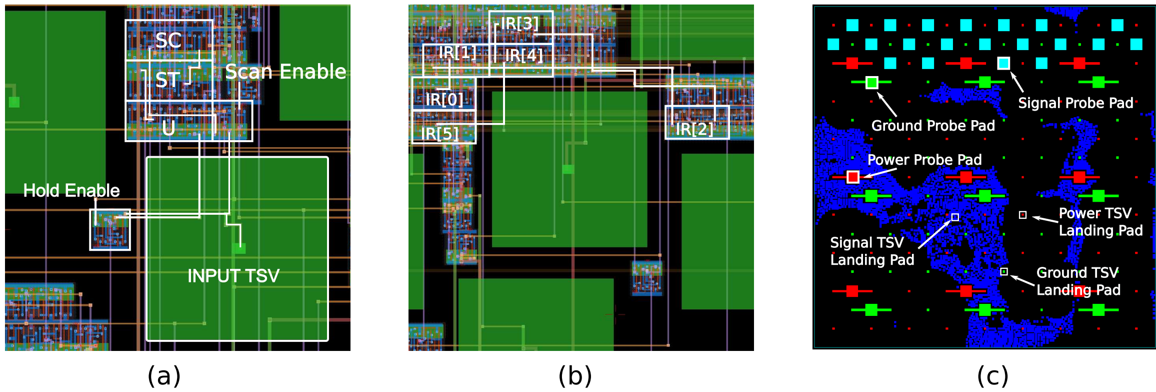


Figure 17: GDSII images. (a) A close up of a TSV and its WBR, (b) IEEE 1500 Instruction Register Chain, (c) zoom out shot of the top metal layer of the top die, showing TSV landing pads and probe pads

Two designs are picked from the OpenCores benchmark suite and implemented in two dies. Design statistics are shown in Table 2. This table splits up the statistics on a die by die basis. The top die does not have any TSVs, and hence that particular entry is blank. This table also reports the results of ATPG for both stuck-at faults as well as transition faults.

In all the following experiments, each die is assumed to have five scan chains. Since the power consumption of stuck at tests can be controlled by reducing the frequency, all power numbers and IR-drop results focus on transition tests. Five transition test vectors are picked from each die, and used as representative vectors. The test vectors of the bottom die are prefixed with “BD” , and those of the top die with “TD”. Since ATPG runs in a greedy fashion, the first few vectors test a larger number of faults per vector than later vectors.

Table 2: Design Statistics for two designs, split by die.

	Jpeg		FFT	
	Bottom Die	Top Die	Bottom Die	Top Die
Gate Count	214,641	197,187	328,512	296,929
# Scan F.F	15,828	22,219	87,681	78,503
# Signal TSV	2,164	-	2,879	-
S-A Coverage (%)	99.77	99.61	99.99	99.99
S-A Patterns	2012	2217	12180	11610
Tr Coverage (%)	98.93	97.74	99.92	99.90
Tr Patterns	3892	5200	61,798	55,656

Therefore, choosing five vectors at random out of the first few generated gives patterns with high switching activity. Since only a single die at a time is tested in the following experiments, the clocks to all the scan flip flops of the die not being tested are gated off, which helps reduce power consumption.

2.2.4.1 Overhead Study

This section discusses the overhead involved in adding the IEEE 1500 wrappers to different designs. The overhead is computed with respect to wirelength, gate count, area, and power, and plotted in Figure 18.

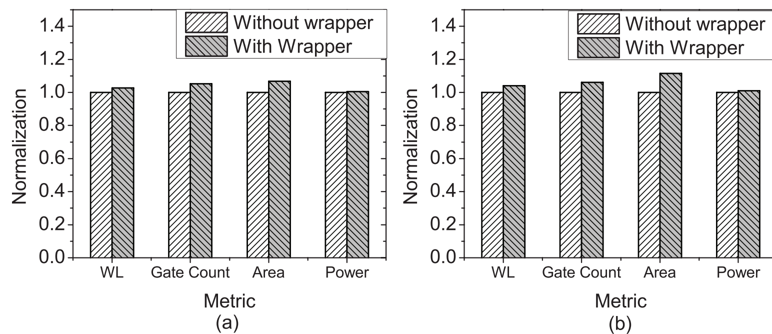


Figure 18: Various overheads involved in adding wrappers for (a) FFT and (b) Jpeg

From this graph, it is observed that there is around a 10% increase in gate area for jpeg, but this reduces to 5% in the case of FFT. This is because FFT has a smaller TSV to gate ratio. For both designs, the wirelength and gate count increase by less than 5%. In addition, only a small increase in the power consumption is observed in both circuits. This because

the test related elements do not switch during the normal operation, and any power increase comes only from the small increase in the wirelength.

2.2.4.2 *Test Time Study*

This section discusses the change in test time for different test types and test configurations. A summary of results is shown in Table 3.

Table 3: Post-bond test time results. All test times are in cycles

Design	Die	Stuck-at test			Transition test		
		[44] ($\times 10^6$)	This Work ($\times 10^6$)	% Inc.	w/o TSV ($\times 10^6$)	with TSV ($\times 10^6$)	% Inc.
FFT	Bot.	220.6	227.6	3.17	1155.0	-	-
	Top	189.0	195.7	3.53	938.2	1002.0	6.83
Jpeg	Bot.	7.2	8.1	12.02	15.7	-	-
	Top	10.8	11.8	8.87	27.6	32.1	16.29

The test time is reported for post-bond test only, as the number of vectors is identical in the pre-bond case. The third and fourth column refers to the test time obtained by running stuck at tests only. The test times are compared against [44], which implements a stuck-at architecture only. Since the proposed architecture has has one additional flip-flop per WBR, the test time is expected to increase. It is observed that this increase reduces with an increase in the circuit size. Columns 6 and 7 compare test times of the top die, when tested through its own WBR, as opposed to through that of the bottom die. This corresponds to testing of the top die without, and with TSV test. Since the latter case has a longer chain length, the test time increases. Again, this increase is observed to be proportional to the circuit size. If this increase is found to be unacceptable, the WBR chain in the top die can be bypassed, incurring some additional area and wirelength costs due to extra multiplexers.

2.2.4.3 *Power Study*

This section evaluates the change in power consumption from pre-bond to post-bond test, as well as across different test patterns. In the case of the top die, post-bond without and with TSV test is also compared. These results are plotted in Figure 19.

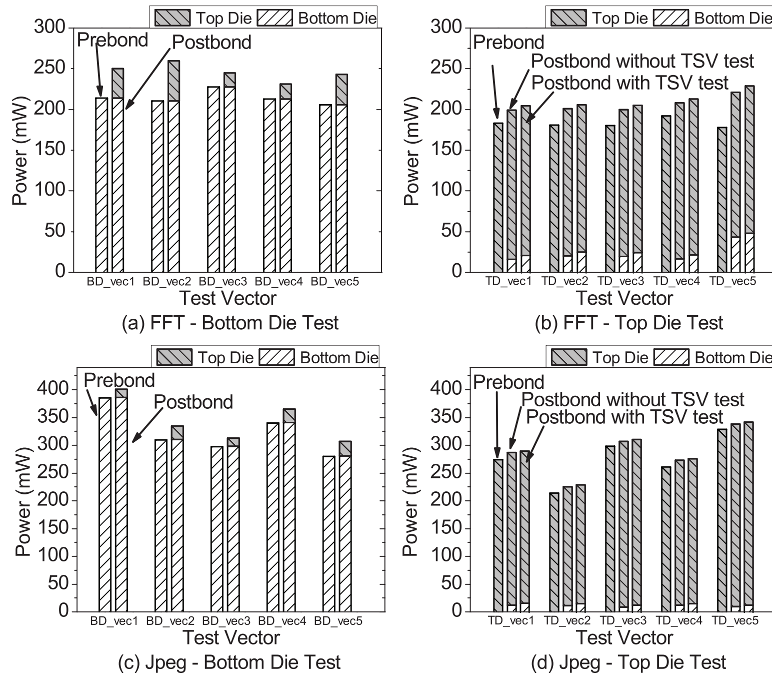


Figure 19: Total Power comparison among (1) pre-bond, (2) post-bond without TSV test, and (3) post-bond with TSV test under five different test vectors.

The total power consumed in each case is split into the contribution by each die. From these graphs, it is observed that the power consumed by a particular die changes very little when moving from pre-bond to post-bond test. However, the other die consumes some additional power due to leakage and switching in the test circuitry, leading to an increase in the overall power. Furthermore, when the top die is tested in conjunction with TSVs, the power consumed by both dies increase, compared to the case when TSVs are not tested. This is because the logic driving TSVs in each die now consumes more power.

2.2.4.4 Pre-bond IR-drop

Here, the impact of different configurations of power probe pads on the voltage drop during the pre-bond test is studied. Since the bottom die receives power from solder bumps, it is of no interest in this study, and hence results focus on the top die only. As mentioned earlier, the probe pads are placed in a regular grid like fashion, at different pitches, and different configurations. The results are shown in Figure 20.

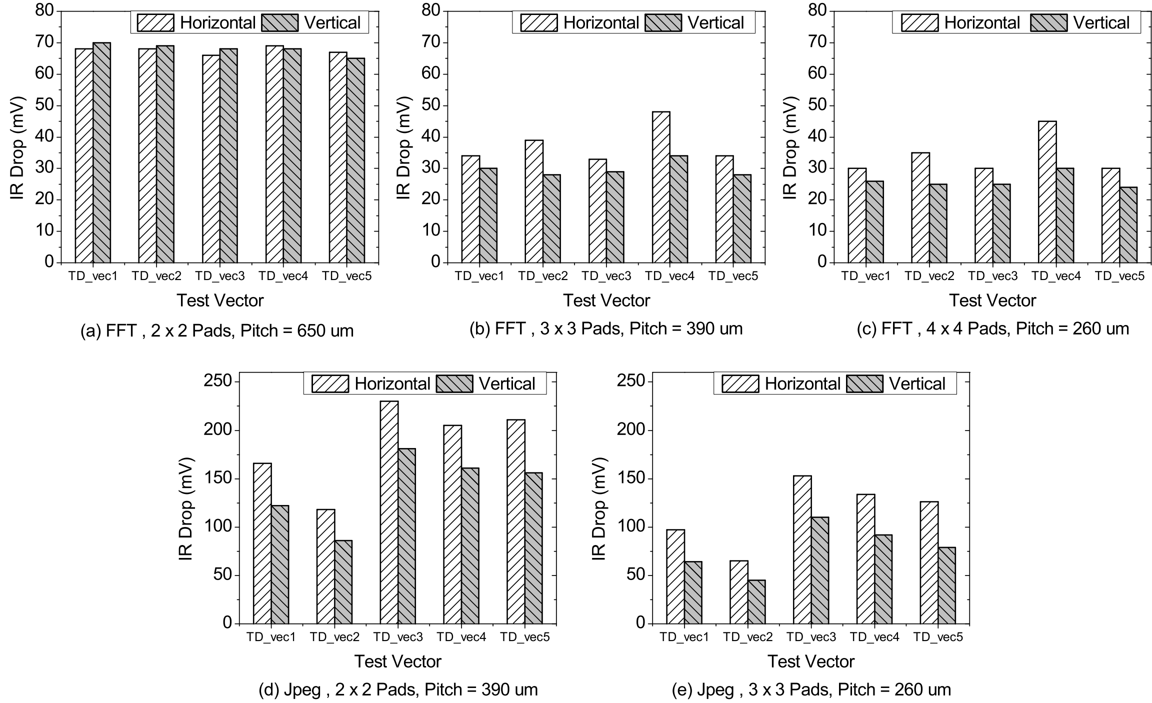


Figure 20: Pre-bond IR-drop under different probe pad configurations and test vectors for FFT (a, b, c) and Jpeg (d, e)

As expected, the IR-drop goes down if the pitch of probe pads go down. It is interesting to note that the vertical configuration almost always outperforms the horizontal configuration. This is because the standard cells receive power from horizontal metal stripes, and placing pads in a horizontal configuration would simply mean that the same stripes get power at two locations. However, in the vertical configuration, more of these stripes will get a direct connection to power, and hence the IR-drop reduces.

As observed for the 2×2 configuration of probe pads of the circuit jpeg, the IR-drop can be quite high. One obvious solution would be to go back to ATPG, and constrain the power budget. This would increase the total number of vectors, and hence the test time. Instead, this project investigates whether any improvement in the IR-drop can be achieved by cleverly placing probe pads. A manually optimized configuration, along with IR-drop maps are shown in Figure 21. Therefore, a careful choice of probe pad locations can reduce the IR-drop.

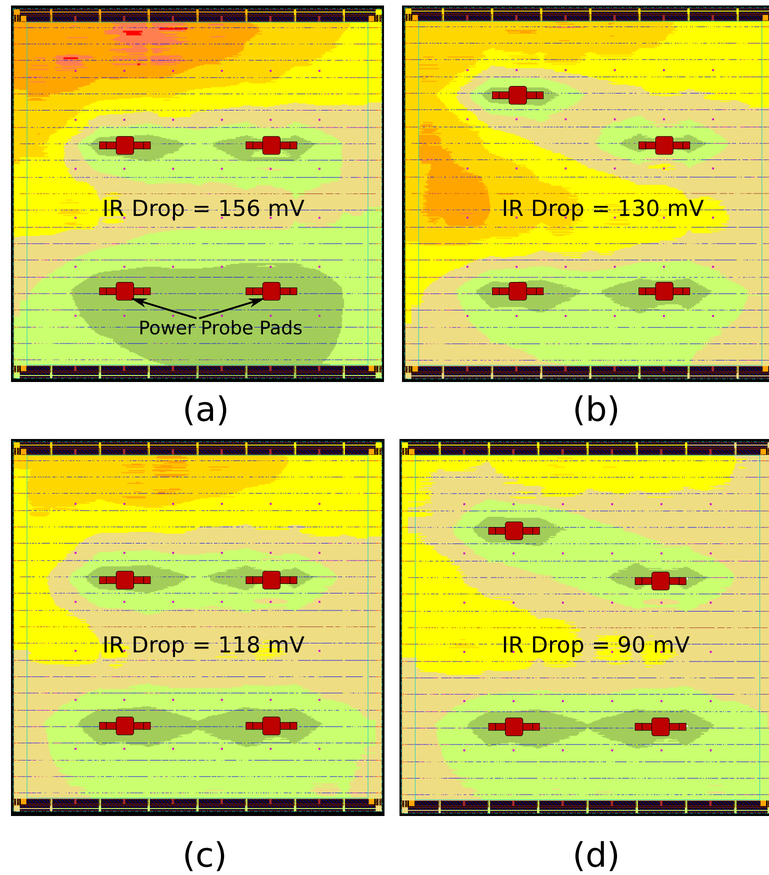


Figure 21: IR-drop maps before (= a, c) and after (= b, d) probe pad optimization.

2.2.4.5 Pre-bond vs. Post-bond IR-drop

This section studies how the voltage drop of a particular die changes depending on the stage in the bonding process. These results are plotted in Figure 22. In the case of the top die, the lowest pre-bond voltage drop achieved among all possible combinations is plotted. Not surprisingly, the post-bond IR-drop of the top die is much lower than the pre-bond case. This is because in the post-bond case, the top die receives power through TSVs at a much finer pitch than the probe pads in the pre-bond case. The small increase in the power consumption, when tested with TSVs is not sufficient to cause any change in the IR-drop. It is interesting to note however, that the IR-drop of the bottom die also reduces slightly during post-bond test, even though it still receives power from the same locations, and has a slightly higher power consumption. This is because during the post-bond test of the bottom

die, the top die consumes very little power, yet attaches its entire power grid in parallel to that of the bottom die. This reduces the equivalent resistance of the power grid, and hence the IR-drop is lower.

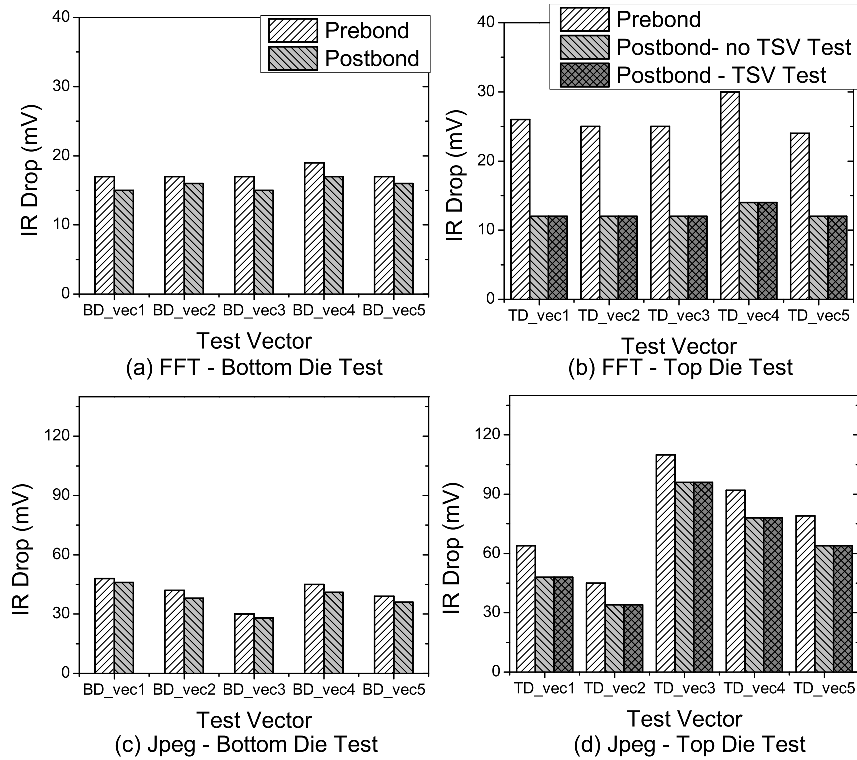


Figure 22: Comparison between pre-bond and post-bond IR-drop. (a) FFT, bottom die, (b) FFT, top die, (c) Jpeg bottom die, (d) Jpeg, top die

2.2.4.6 *Normal vs. Test Mode*

Since transition fault testing aims to switch as many nets as possible with one vector, the IR-drop during the test mode is expected to be much higher than the IR-drop during the normal mode. The normal mode IR-drop of Jpeg was found to be $10mV$, and that of FFT was found to be $6mV$. When compared with the post-bond numbers from Figure 22, it is clear that test mode has much higher IR-drop.

2.3 Test-time Estimation for 3D ICs

During early design space exploration, a large number of possible partitioning solutions are evaluated w.r.t. power, performance, area, TSV count, etc. The TSV count includes the number of signal TSVs, as well as estimates of TSVs for power delivery, clock, thermal, and test. The number of test-TSVs depend on the test architecture, and includes TSVs required for control, as well as those required to pump data. If test-TSVs are not accounted for during partition evaluation, downstream design steps may have insufficient area to add these TSVs. One such example is shown in Figure 23, where floorplanning was carried out considering only signal TSV count. Insufficient area remains to add other TSVs such as clock, power and test. The only solution is to expand die area, which increases cost, and reduces yield. To avoid this, test-TSVs need to be accounted for during the partitioning process.

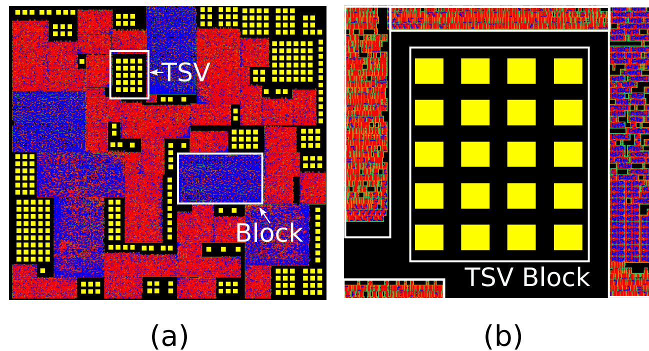


Figure 23: (a) GDSII screen shot of a single die of a block-level 3D IC (b) Zoom in shot of the boxed TSV block in (a)

The chosen test architecture determines the number of control test-TSVs, while the number of TSVs required to pump data is variable, and left up to the design engineer. Only the latter is of interest, as the former remains constant irrespective of partition. In the remainder of this section, test-TSVs refer only to those TSVs used to carry test vectors and responses, and control test-TSVs can be treated as a separate, fixed constant.

If a fixed number of test-TSVs ($TSV_{t,f}$) are allocated during partitioning, there is the possibility of overestimating the real total TSV count of a partition. It has been shown [50]

that pareto-optimality exists in the test-TSV count. If $TSV_{t,po}$ is the pareto-optimal number of test-TSVs, any TSVs allocated beyond this will not yield a reduction in test time. The actual number of test-TSVs used during scheduling is given by

$$TSV_t = \min(TSV_{t,f}, TSV_{t,po}) \tag{2}$$

In area critical designs, when $TSV_{t,f}$ is small, it is usually the smaller of the two, so it serves as a reasonable estimate. However, if $TSV_{t,f}$ is large, and it was used as an estimate for TSV_t, several candidate partitioning solutions would be discarded for having too many TSVs. Therefore, an accurate estimate of $TSV_{t,po}$ is required, and it needs to be quickly computed to be incorporated into automatic partitioning.

Existing test-scheduling algorithms such as [49, 50] focus on determining the test time given a fixed test-pin and test-TSV constraint [49]. Using such algorithms to determine $TSV_{t,po}$ would require repeatedly applying them for different test-TSV constraints, and finding the point where there is no reduction in test time. While this process will work if the partition is fixed, it is too slow to be used during early design space exploration. In this section, a fast and accurate estimate of the pareto-optimal number of test-TSVs required for a given 3D partition is derived. Since the test time estimate is meant to be used during design space exploration, block-level designs are assumed, where the blocks are all soft, and top-level interconnect tests are ignored. To validate results, the ILP-based test scheduling algorithm presented in [49] is used to compute test time for a given partition.

2.3.1 Die-level partitioning

Die-level partitioning is studied first, where partitioning implies die ordering. While the solution space is small, and exhaustive search methods can easily be applied, insights gained in this section are used to explain block-level partitioning later.

2.3.1.1 Two-die stack

A two tier die-level stack is the simplest form of a 3D IC, and there are only two partitions possible. Furthermore, only two test scheduling options exist, serial or parallel test. In serial test, each die is tested one at a time, the bottom die with all the test-pins, and the top die with all the test-TSVs. In parallel test, the test-pins are divided between the bottom and the top die. Three circuits are considered, and shown in Figure 24. The first circuit is a homogeneous stack, and the next two are different die-level partitions of a heterogeneous stack. Each die is a circuit taken from the ITC'02 SOC benchmarks [45].

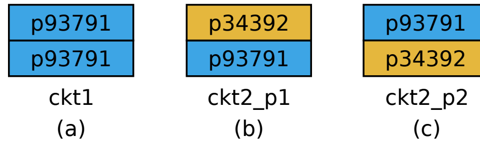


Figure 24: Three different circuits considered for die-level partitioning of a two-die stack. (a) A homogeneous stack, (b & c) Two different partitions of a heterogeneous stack. A larger number implies the die is more complex.

Since the solution space is small, all possible test scheduling options are tried, and the pareto-optimal TSV count for both serial and parallel test is tabulated in Table 4. Fifty test-pins are assumed, and the test-TSV count is swept to obtain the minimum test time and $TSV_{t,po}$. The parallel schedule offers lower test time, and would be chosen by any test scheduling algorithm. For the homogeneous stack, an equal division of test-pins is optimal, which implies that $TSV_{t,po}$ is half of the number of test-pins, or 25. For the heterogeneous stack however, it is observed that both partitioning options give the same minimum test time, but $TSV_{t,po}$ is different. As expected, the partition with the more complex die on top requires more test-TSVs to obtain minimum test time.

2.3.1.2 Multi-die stack

This section tabulates the test time for a given set of partitions under fixed test-pin and TSV constraints, and then uses this information to identify the characteristics of the partition that affects the test time. The different multi-die circuits considered are shown in Figure 25.

Table 4: The optimal test times (in cycles) achieved for a two-die circuit, along with the TSV usage at which this optimum time is reached.

Circuit	Serial Test		Parallel Test	
	T_{min}	$TSV_{t,po}$	T_{min}	$TSV_{t,po}$
ckt1	2,447,767	47	2,363,730	25
ckt2_p1	1,931,750	47	**1,899,170**	**19**
ckt2_p2	1,940,656	47	**1,899,170**	**31**

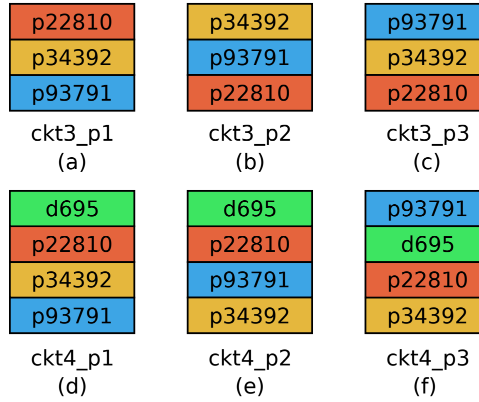


Figure 25: Circuits considered for die-level partitioning of multi-die stacks. (a - c) three die stack, (d - f) four die stack. A larger number implies the die is more complex.

TSV constraints can be assigned in two ways. The first method is *uniform TSV constraints*, which allocates an equal TSV budget to all the dies. The second method is *tapering TSV constraints*, which allocates more TSVs for the lower dies (closer to the package), and less TSVs for the upper dies. The test time is computed using ILP-based scheduling. The test time difference for both types of constraints is studied, and tabulated for three and four dies in Tables 5 and 6, respectively.

Table 5: The test times for die-level partitioning of a three-die 3D IC, considering both uniform and tapered TSV constraints.

P_{max}	TSV_{max}		Test time (cycles)		
	D2-D1	D3-D2	ckt3_p1	ckt3_p2	ckt3_p3
50	50	50	**2,197,060**	**2,197,060**	**2,197,060**
	30	30	2,252,535	**3,138,753**	**3,138,753**
	30	10	2,252,535	3,826,504	7,021,398
70	70	70	**1,541,308**	**1,541,308**	**1,541,308**
	30	30	1,753,753	**3,138,753**	**3,138,753**
	30	10	2,249,017	3,826,504	7,021,398

Table 6: The test times for die-level partitioning of a four-die 3D IC, considering both uniform and tapered TSV constraints.

P_{max}	TSV_{max}			Test time (cycles)		
	D2-D1	D3-D2	D4-D3	ckt4_p1	ckt4_p2	ckt4_p3
50	50	50	50	**2,225,765**	**2,225,765**	**2,225,765**
	30	30	30	2,300,851	**2,597,776**	**2,597,776**
	30	20	10	2,418,438	2,971,786	7,021,398
70	70	70	70	**1,561,751**	**1,561,751**	**1,561,751**
	30	30	30	1,802,068	**2,597,776**	**2,597,776**
	30	20	10	1,919,655	2,971,786	7,021,398

It is clear from these tables that, as expected, the test time of a partition with the most complex dies closest to the package is least. However, under uniform TSV constraints, the test time changes only when the bottom die changes. Any permutation of the upper dies without changing the bottom die does not affect the test time. Furthermore, if the pin and TSV constraints are equal, partitioning has no impact on the test time. If two partitions have the same test time when tested with the same number of TSVs, it follows that they both also have the same $TSV_{t,po}$. This implies that $TSV_{t,po}$ only needs to be updated if the complexity of the bottom die changes. These results are not restricted to these particular simulation settings, and a formal proof is given below.

Lemma 1. *Assume that TSV_{max} is a uniform TSV constraint to test the set of dies D. Let $D_p \subseteq D$ be a subset of the dies tested in parallel within a single test session. Let $p_d = (p_1, \cdots, p_{|D_p|})$ be a division of pins within this test session. If two dies D_i and D_j, $i \neq j \neq 1$ are swapped, then p'_d obtained from p_d by swapping p_i and p_j does not violate P_{max} and TSV_{max} constraints.*

Proof. The number of TSVs in die k (TSV_k) satisfies

$$TSV_k = \max_{l=k}^{|D|} \sum_{m=l}^{|D|} p_m \leq TSV_{max} \ \ \forall k > 1 \tag{3}$$

Since the set of dies D is known to be tested with p_d, Equation (3) is satisfied. It needs to be proved that this equation is also satisfied if D' is tested with $p'd$. Clearly, the greatest

term in Equation (3) occurs when $k = 2$, or at the die immediately above the bottom die. Therefore $\sum_{m=2}^{|D|} p_m$ satisfies the TSV_{max} constraint. If D' is tested with p'_d, this sum does not change, and therefore p'_d also satisfies the TSV_{max} constraint. □

This lemma proves that if two dies are tested in parallel, and then interchanged in the stack, they can still be tested in parallel with the same division of pins. It does not claim that the same old division of pins will be optimal for the new partitioning, just that it is possible without violating TSV and pin constraints.

Lemma 2. *If the set of dies D is tested with a certain test schedule (with uniform TSV_{max} constraints), then any different partition D' with the same bottom die D_1, can be tested with the same test schedule.*

Proof. A test schedule is merely a series of test sessions with dies tested in parallel within the same test session. Since TSVs are multiplexed between two different sessions, it is enough to show that a single test session can be repeated for D'. From the previous lemma, the test session can be repeated for a different partition with two dies interchanged. It is clear that D' can be obtained from D with a series of two die exchanges. Therefore D' can also be tested with the same test schedule. □

Again, this lemma does not claim that the same test schedule is optimal for the new partition, but simply that it is possible. Finally, it is proved that the test time is independent of the partition of upper dies.

Theorem 1. *All partitions of a set of dies D with same bottom die D_1 have the same test time under a uniform TSV_{max} constraint.*

Proof. Let D_{all} be the set of all partitions of D with the same bottom die D_1. Using identical TSV_{max} constraints, find the partition with the minimum test time, say D_{min}. Then, from the previous lemma, any other partition $D' \in D_{all}$ can be tested with the same test schedule as D_{min}, and hence also has minimum test time. □

Tables 5 and 6 also show that if the number of test pins is equal to the number of test TSVs, then all partitioning results have the same test time. The proof of this follows from the fact that if $P_{max} = TSV_{max}$, lemma 1 holds for interchanging any two dies, including the bottom die.

2.3.2 Block-level partitioning

Block-level partitioning is the more general case of die-level partitioning. This section studies the change in test time for different partitions under fixed test-TSV constraints, derives lower bounds on the test time, and uses these lower bounds to derive equations for $TSV_{t,po}$. This section assumes uniform TSV constraints.

2.3.2.1 Two-die stack

Ckt2_p2 is taken as a starting solution, and modules are moved across the tiers. Each move results in a new partition. Two types of module moves are performed. The first is moving a module from one die to another, and the other is swapping two modules from different dies. A total of 1000 moves are performed, and test scheduling is carried out for each partition assuming 50 test-pins and different TSV constraints. The results are plotted in Figure 26.

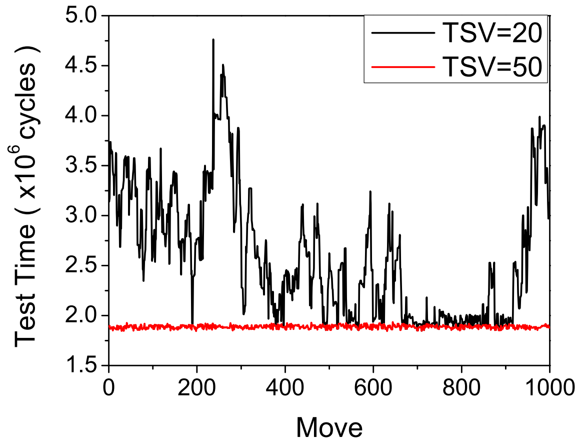


Figure 26: The variation in test time observed for a two-die stack starting with ckt2_p2 and performing 1000 different random moves. 50 test-pins and 2 different test-TSV constraints are assumed.

As observed in the previous section, if the test-TSV constraint is high enough, all partitions have similar test time. With lower test-TSV constraints (= 20), it is observed that a significant number of partitions have much higher test time, indicating that their $TSV_{t,po}$ is higher. There are also partitions, however (Moves 650-800), that have close to the minimum test time, indicating that their $TSV_{t,po}$ is close to 20. These results are explained on the basis of the lower bounds derived below.

Lower bound on test time For a module m, let i_m, o_m, and b_m be the number of input, output, and bi-directional ports, respectively. Further, let p_m be the number of patterns required to test that module. Let f_m be the number of flip flops in that module. In the case of hard modules, f_m is simply the sum of the lengths of the internal scan chains. The number of stimulus (ts_m) bits is the sum of i_m, b_m, and f_m, and the number of response bits (tr_m) is the sum of o_m, b_m, and f_m. The complexity of a module m is then defined as:

$$c_m = \max(ts_m, tr_m) \cdot pm + min(ts_m, tr_m) \tag{4}$$

Note that this is simply the test data volume of that particular module, neglecting the one cycle required to run the test. Given a set of modules M, the complexity of that set C_M is defined as the sum of the complexities of all its constituent modules i.e., $\sum_{m \in M} c_m$. Although similar to the ITC'02 [45] definition of complexity, this formulation is linear. This implies that irrespective of any partition of the modules M into M_1 and M_2, the sum of C_{M_1} and C_{M_2} will always result in C_M.

Given a set of modules M and P pins with which to test them, a lower bound on the test time of a 2D design based on the amount of data that needs to be pumped into it was given by [17], and can be re-written as:

$$LB_{2D}(M, P) = \left\lceil \sum_{m=1}^{|M|} \frac{c_m}{\lfloor P/2 \rfloor} - \sum_{m=1}^{|M|} \frac{\min(tr_m, ts_{m+1})}{\lfloor P/2 \rfloor} \right\rceil + \min_{m=1}^{|M|} p_m \tag{5}$$

Let M_{3D} be the set of all modules in the 3D stack. M_1 is the set of modules in the bottom die, and M_2 the set of modules in the top die. Let LB_{M_i} denote the lower bound

of the test time of the set of modules M_i. First, the lower bounds induced by both TSV and pin constraints are considered. It is assumed that $TSV_{max} <= P_{max}$, as any additional TSVs will simply be wasted. The maximum test-pins available to the bottom and top dies are P_{max} and TSV_{max}, respectively. Therefore, a partition-dependant lower bound is given by:

$$LB_{dep} = \max\{LB_{2D}(M_1, P_{max}), LB_{2D}(M_2, TSV_{max})\} \tag{6}$$

This lower bound can be improved by considering that every module in the 3D stack can be tested with no more than P_{max} pins. Such a lower bound is partition independent, and is given by:

$$LB_{indep} = LB_{2D}(M_{3D}, P_{max}) \tag{7}$$

This lower bound holds irrespective of the partition or the TSV count. The overall lower bound is then given by the maximum of the partition independent and dependent lower bounds, and it can be reduced to:

$$LB_{3D} = \max\{LB_{2D}(M_{3D}, P_{max}), LB_{2D}(M_2, TSV_{max})\} \tag{8}$$

Once the lower bound is defined, its behaviour w.r.t. change in the partitioning solution needs to be captured. A partition-dependent metric, the complexity factor (CF) for a two-die stack is defined as:

$$CF = \frac{C_{M_1}}{C_{M_1} + C_{M_2}} = 1 - \frac{C_{M_2}}{C_{M_1} + C_{M_2}} \tag{9}$$

Varying CF from 0 to 1 captures all types of partitions. A CF of 0 means that all modules are in the top die, and a CF of 1 means that all modules are in the bottom die. There exists a CF beyond which the lower bound becomes constant, as proved below.

Theorem 2. *$LB_{2D}(M_2, TSV_{max})$ decreases with increasing CF, and intersects LB_{indep} for all values of $TSV_{max} < P_{max}$.*

Proof. The first statement is trivial. If CF increases, it implies that C_{M_2} reduces, and this will reduce the lower bound on M_2. Next, when $CF = 0$, all the modules are in Die 2,

M_{top} becomes M_{3D}. Since $TSV_{max} < P_{max}$, $LB_{2D}(M_{3D}, TSV_{max}) > LB_{2D}(M_{3D}, P_{max})$. When $CF = 1$, the top die is empty with lower bound zero, and therefore, $LB_{2D}(M_{top}, TSV_{max}) < LB_{indep}$. This shows that somewhere in between a CF of 0 and 1, they intersect. □

To calculate the value of this threshold, a linear approximation of Equation (8) is developed. It is assumed that the scan unload and scan load of successive modules are not overlapped. In addition, the third term in Equation (5) is neglected, as it is small when compared with the first. Equation (8) can then be approximated as:

$$LB'_{2D}(M, P) \approx 2 \cdot C_M / P \tag{10}$$

The lower bound then becomes

$$LB'_{3D} = 2 \cdot \max\left(\frac{C_{M_{3D}}}{P_{max}}, \frac{C_{M_2}}{TSV_{max}}\right) \tag{11}$$

The threshold complexity factor is the complexity factor when both terms are equal, and beyond which test time does not change. It is given by

$$CF_{th} = 1 - \frac{TSV_{max}}{P_{max}} \tag{12}$$

Note that this threshold value only depends on the TSV and pin constraints and not on the actual design or partition.

With these simplifications, the approximate lower bound on the 3D test time can be written as

$$LB'_{3D} = 2C_{M_{3D}} \times \begin{cases} (1 - CF)/TSV_{max} & 0 \leq CF \leq CF_{th} \\ 1/P_{max} & CF_{th} \leq CF \leq 1 \end{cases} \tag{13}$$

This gives a linear model for the lower bound, with both design dependant and independent terms. The shape of the lower bound curve is independent of design, and it is simply shifted up or down depending on the particular design. This linear model gives a way to predict the

lower bounds on the test time without having any real partition information. The converse of Equation (12) can be used to find out the pareto-optimal number of TSVs for a given partition. Given a partition P with complexity factor CF_P, $TSV_{t,po}$ can be written as:

$$TSV_{t,po} = P_{max} \times (1 - CF_P) \tag{14}$$

This equation essentially finds the TSV count for which this partition is at the threshold complexity factor. Increasing the TSV count beyond this value implies that the first term in Equation (11) is greater than the second term, and since it is a constant, the test time does not reduce. This is the definition of $TSV_{t,po}$.

Test time vs. lower bound This section plots the test time versus the CF, to observe how different partitions affect the test time. In addition, the approximate lower bound is plotted on the same scale to investigate how the test time curve compares to the lower bound curve, and is shown in Figure 27.

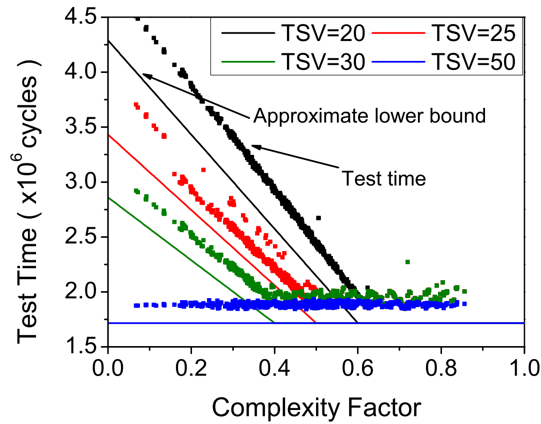


Figure 27: Comparison between the measured test time and approximate lower bound of test time (= Equation 13) for a 2 die stack. The number of test pins is 50.

As expected, the test time curve follows the general shape of the lower bound, but is shifted upwards by some amount. Most importantly, the threshold complexity factor CF_{th} for both the test time and the lower bounds is similar. Therefore, the lower bound gives

the designer a very good estimate of what the shape of the test time curve is. Therefore, $TSV_{t,po}$ is well estimated by Equation (14).

2.3.2.2 Multi-die stack

Similar to the experiment done with two dies, ckt3_p1 is used as the initial design. Then, 1000 random moves are made and the variation in test time is observed. Although the moves random, specific kinds of moves are made. The first $1/3$ moves are performed only between Die 1 and Die 2. The next $1/3$ are only between Die 1 and Die 3. The third and final $1/3$ is made between Die 2 and Die 3. The test time is computed using ILP with a test-pin constraint of 50 and 2 different uniform TSV constraints and the results obtained are plotted in Figure 28.

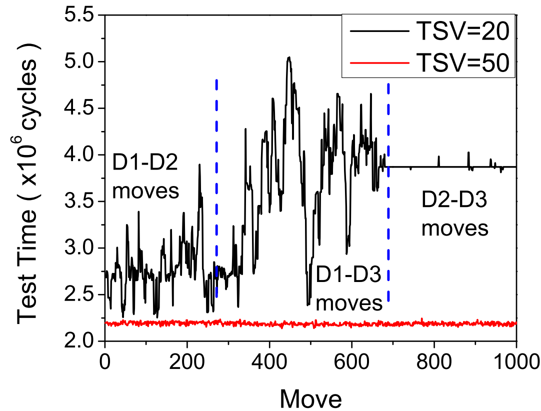


Figure 28: Variation in test time observed while performing 1000 random moves, starting with ckt3_p1. The test time is computed assuming 50 test-pins, and 2 different uniform TSV constraints (20 vs 50 per-die).

From these results, it is again observed that if sufficient TSVs are available, the test time does not vary much, indicating that all partitions have at least $TSV_{t,po}$ TSVs. If, however, sufficient TSVs are not available, there is significant variation in the test time. Most interestingly however, similar to the die-level partitioning, moves between the upper dies do not change the test time. These results are explained on the basis of lower bounds on test time, as described next.

Lower bound on test time This section generalizes the results obtained for the two-tier case. The lower bound on the top die can be written as:

$$LB_{M_{|D|}} = LB_{2D}(M_{|D|}, TSV_{max,|D|}) \tag{15}$$

For the die $|D| - 1$, the lower bound can be written as

$$LB_{M_{|D|-1}} = LB_{2D}(M_{|D|-1}, TSV_{max,|D|-1}) \tag{16}$$

Now, all the modules in the upper two dies can be tested with at most $TSV_{max,|D|-1}$ TSVs. Therefore

$$LB_{M_{|D|,|D|-1}} = LB_{2D}(M_{|D|} \cup M_{|D|-1}, TSV_{max,|D|-1}) \tag{17}$$

The true lower bound on the test time of the upper two dies is simply the maximum of Equations (15), (16), and (17). Similar lower bounds on all dies can be obtained by inductively working backwards from the top die. The lower bound of test time to test all upper tiers can then be written as

$$LB_{D-D_1} = \max_{i=2}^{|D|}\{LB_{2D}(\cup_{j=i}^{|D|} M_j, TSV_{max,i})\} \tag{18}$$

This is the time to test the upper die with $TSV_{max,|D|}$ TSVs, the upper two dies with $TSV_{max,|D|-1}$ and so on. The test time of the entire 3D stack can than be given by.

$$LB_{3D} = \max(LB_{3D-D_1}, LB_{2D}(M_{3D}, P_{max})) \tag{19}$$

This is a general equation, for arbitrary TSV constraints. However, for the special case when all the TSV constraints are equal, say TSV_{max}, this can be reduced to

$$LB_{3D,eq} = \max(LB_{2D}(\cup_{i=2}^{|D|} M_i, TSV_{max}), LB_{2D}(M_{3D}, P_{max})) \tag{20}$$

Approximate formulae can then be obtained by linearisation

$$LB'_{3D} = \max\left\{\max_{i=2}^{|D|} \frac{2 \cdot \sum_{j=i}^{|D|} C_{M_i}}{TSV_{max,i}}, \frac{2 \cdot C_{M_{3D}}}{P_{max}}\right\} \tag{21}$$

If uniform TSV constraints, say TSV_{max}, are assumed, then this reduces to

$$LB'_{3D,eq} = \max\left\{\frac{2 \cdot \sum_{j=2}^{|D|} C_{M_i}}{TSV_{max}}, \frac{2 \cdot C_{M_{3D}}}{P_{max}}\right\} \tag{22}$$

This shows that the lower bound is independent of the partition of the upper dies. For uniform TSV constraints, a complexity factor can be defined as

$$CF = \frac{C_{M_1}}{C_{M_{3D}}} = 1 - \frac{\sum_{j=2}^{|D|} C_{M_i}}{TSV_{max}} \tag{23}$$

Note that this CF has a slightly different meaning from that of the two-die case. Here, if CF = 1, then all modules are in the bottom die as usual, but a CF of 0 simply means that no modules exist in the bottom die. With this definition of CF, the definitions of the threshold complexity factor CF_{th}, and $TSV_{t,po}$ are identical to the two-die case.

Test time vs. lower bound The test time vs CF for a four-die circuit is plotted by starting with ckt4_p1, and performing 1000 different moves. The test-pin constraint is assumed to be 100, and a uniform TSV constraint is assumed. The TSV numbers are chosen such that the TSV-to-pin ratio is the same as that of the two-die case. This would imply that the shape of the approximate lower bounds is exactly the same, but the curve will have a different magnitude. The purpose of this is to demonstrate that different circuits tested under the same TSV-to-pin ratio indeed have similar test time curves. This is plotted in Figure 29.

As observed from this figure, the slope of the test time curve as well as the threshold complexity values are dependent only on the TSV and pin constraints, and not on the circuit being tested. This implies that Equation (14) gives us a good estimate of $TSV_{t,po}$, even for more than two tiers.

2.3.3 Case Studies

In this section, benchmark circuits are chosen from the IWLS'05 benchmark suite, and the developed theory is applied to it. Two circuits are chosen, and details are tabulated in

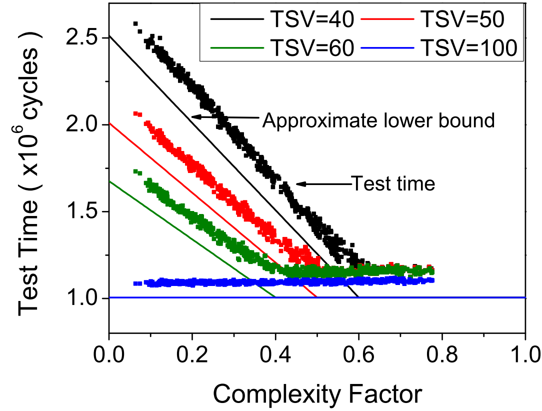


Figure 29: Comparison between the measured test time and approximate lower bound for a four-die stack. The test pin constraint is assumed to be 100.

Table 7. ATPG for each module is performed using Synopsys Tetramax, and this table lists the average and standard deviation of test data volume (TDV) among all modules. Uniform TSV constraints are assumed in all experiments involving more than two dies.

Table 7: Details of benchmark circuits used, showing the average and standard deviation of the test data volume among all modules.

Circuit	#Modules	Average TDV	Std.Dev TDV
b19	57	141,489	168,833
des_perf	51	18,820	18,857

2.3.3.1 Test time variation

The objective of this experiment is to confirm that different partitions with the same bottom die have similar test time. This will justify the definition of complexity factor, which in turn translates to a more accurate $TSV_{t,po}$. Four die implementations of the two benchmarks are taken as the baseline, and 500 moves that change the complexity of the bottom die are performed. Next, an additional 500 moves that change the complexity of the upper dies, but maintain the bottom die constant are performed. The variation observed for each type of move is plotted in Figure 30. The variation is computed as $(t_{max} - t_{min})/t_{min}$, where t_{max} and t_{min} are the maximum and minimum test times respectively.

It is observed that moves involving the bottom die have significantly higher variation

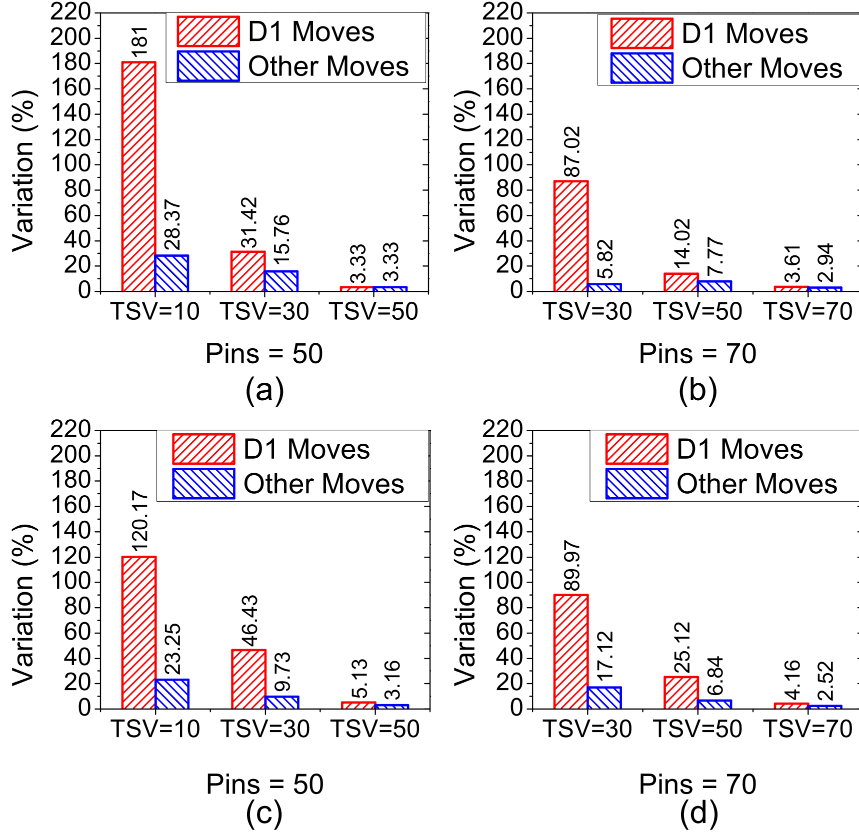


Figure 30: Comparison of the variation in test time observed between moves involving the bottom die (= D1 moves), and all other moves. The numbers are reported for four-die implementations of (a,b) b19, (c,d) des_perf.

when compared with moves that do not, validating the assumptions made. It is also observed that if the test-TSV constraint is increased, the variation in the moves involving the bottom die decreases. This is because with increased test-TSV constraints, a greater fraction of all possible partitions already meet $TSV_{t,po}$.

2.3.3.2 Threshold complexity factor prediction

Correct prediction of CF_{th} is important, as it directly translates in to the correct prediction of $TSV_{t,po}$. Theoretically, it is computed by Equation (12). This section validates the fact that CF_{th} is independent of design and only depends on the ratio between TSV and pin constraints.

The experimental CF_{th} is computed as follows. One thousand partitions of a design are

taken, and the CF and test time of each one is computed. Bins are created with respect to CF, with a bin size of 0.005. For each bin, the average test time of all the partitions (using ILP) that lie within that bin is computed. The threshold CF is computed as the first bin for which the test time is within 10% of the minimum test time observed for that particular pin and TSV constraint.

The theoretical and experimental results observed are plotted in Figure 31. Different TSV and pin constraints that lead to the same CF_{th} are considered. In addition, both two and four die implementations of both designs are plotted. This figure shows that the theoretical formula does indeed give results close to the experimentally observed ones, which means that CF_{th}, and equivalently $TSV_{t,po}$, can be quickly and accurately estimated.

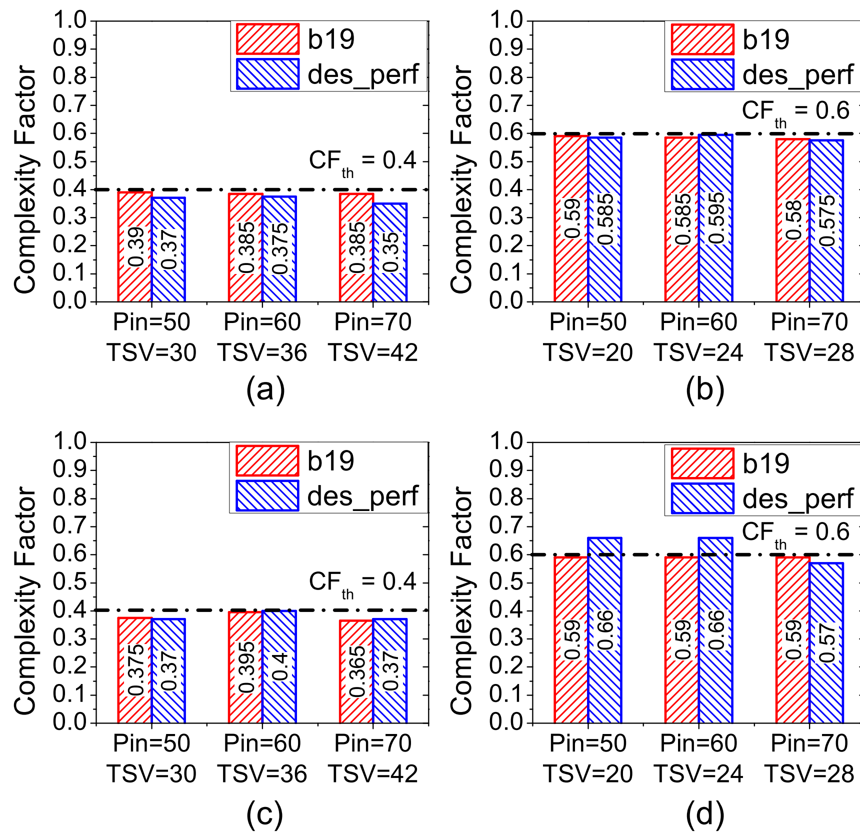


Figure 31: Comparison of theoretical and experimental threshold complexity factors under various TSV and pin constraints. (a,b) Two-die stack, (c,d) Four-die stack.

2.3.3.3 Over-design reduction

In this experiment, $TSV_{t,po}$ is computed during a simulated partitioning process, and its variation is observed. The partitioning process is simulated by taking an initial circuit, and performing 1000 different random moves on it. The results are plotted assuming 50 test-pins in Figure 32.

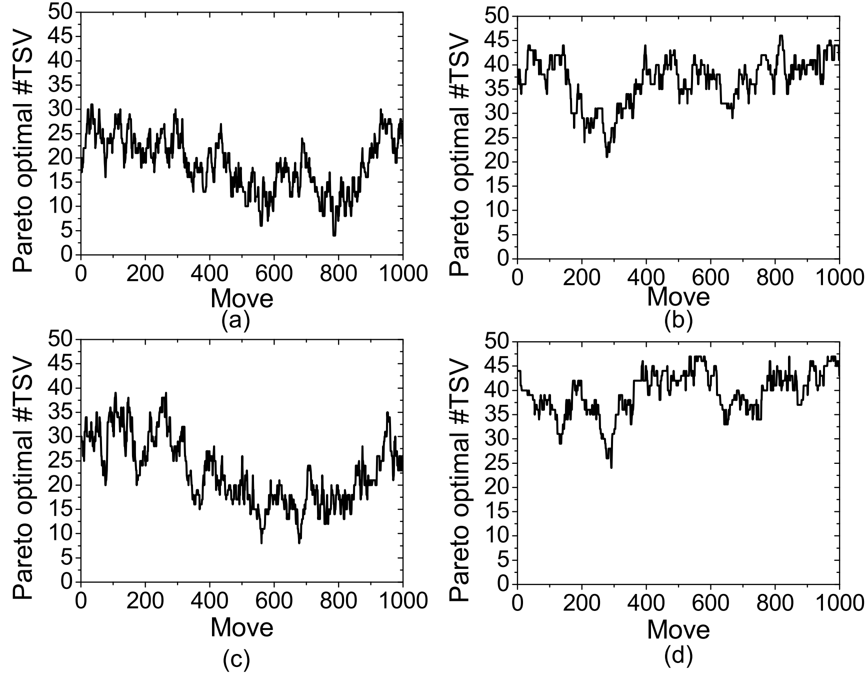


Figure 32: The variation in $TSV_{t,po}$ observed while performing 1000 different random moves, assuming 50 test-pins. (a) b19 two-dies, (b) b19 four-dies, (c) des_perf two-dies and (d) des_perf four-dies.

From this figure, it is observed that if a fixed TSV constraint is used, then there is the possibility of over-design depending on what that constraint is. If it is quite low (e.g., 10), then the $TSV_{t,po}$ is always greater than this, and no resources are wasted. If however the fixed TSV constraint is high (e.g., 40), then the actual number of TSVs required can be much lesser than this, and correct prediction of $TSV_{t,po}$ helps eliminate resource wastage. It is also observed that increasing the number of tiers increases $TSV_{t,po}$. This is expected, as more tiers require more TSVs to test them with minimum test time.

2.4 Summary

This chapter presented various techniques for design-for-test for TSV-based 3D ICs, which is one of the last challenges facing their adoption. First, a technique to create 3D scan chains was developed. Unlike previous approaches, this technique is pre-bond testable. The impact of the number of scan TSVs on the scan wirelength was also presented.

Next, an architecture for transition delay fault testing of 3D ICs was presented. This architecture supports pre-bond as well as post-bond test of the logic, as well as post-bond test of all the TSVs. In addition, since IR-drop is an issue during transition testing, techniques to mitigate IR-drop were presented. In addition, adding probe pads into the layout for pre-bond test access was also discussed.

Finally, this chapter presented techniques to quickly and accurately estimate the test time of a given 3D IC partition. This estimate can be used during the partitioning process to assess the total number of test-TSVs required by a given partition.

CHAPTER III

PHYSICAL DESIGN FOR BLOCK-LEVEL MONOLITHIC 3D ICS

Since re-designing existing logic, memory and IP blocks for 3D incurs significant design overhead and cost, 3D ICs will first focus on reusing existing 2D blocks [29, 61, 31]. These 2D blocks are placed in a 3D space and connected together using MIVs. However, since block-level designs have only a few inter-block wires, this design style is also a prime target for TSV-based 3D ICs. A few works have considered adding TSVs into existing whitespace blocks at the floorplanning stage. Simultaneous buffering and TSV planning was carried out in [20], but the authors reported inaccurate 3D half perimeter wirelength (HPWL) and timing metrics. An improved algorithm was presented in [61], but the same inaccurate HPWL metric was used. Results based on an improved BB-2D-HPWL metric was presented in [31], and the most accurate HPWL metric based on subnets was presented in [29]. However, none of these papers compared the quality of their engine with that of a commercially available tool, or took the obtained floorplans all the way through place and route to obtain GDSII layouts.

This chapter first presents a 3D floorplanning framework that is capable of handling monolithic 3D ICs as well as TSV-based 3D ICs. The quality of the floorplanning results are validated against a commercial tool. It is shown that, even in coarse-grained block-level designs, monolithic 3D can lead to better designs than in TSV-based 3D. Next, due to the fabrication process, some tiers suffer from degraded performance. This chapter models this performance degradation, and provides a floorplanning technique to make designs more resilient to it.

3.1 3D Floorplanning with Monolithic Inter-tier Vias

3.1.1 Problem Formulation and Overview

A general form of the 3D floorplanning problem can be stated as follows: Given the number of desired tiers, and a set of blocks along with their corresponding widths and heights, determine the (x, y, z) locations of each of the blocks and all MIVs/TSVs.

The overall design flow, assuming hard blocks (i.e., have a fixed aspect ratio) is shown in Figure 33. Floorplanning is performed to determine the location of all the blocks assuming the pins are placed at the center. Once the locations of all the blocks are determined, blocks are updated with the locations of the pins, and a refinement step (i.e., PFPR) is performed out to further minimize wirelength. Note that this refinement is unnecessary for soft blocks, as the pin locations are determined based on the floorplanning result. Different via planning engines are used depending on whether TSVs or MIVs need to be inserted. Finally, separate verilog and DEF files are created for each die/tier with the corresponding connectivity information and location of blocks and TSVs/MIVs, respectively. Each of the above steps are further explained in following subsections.

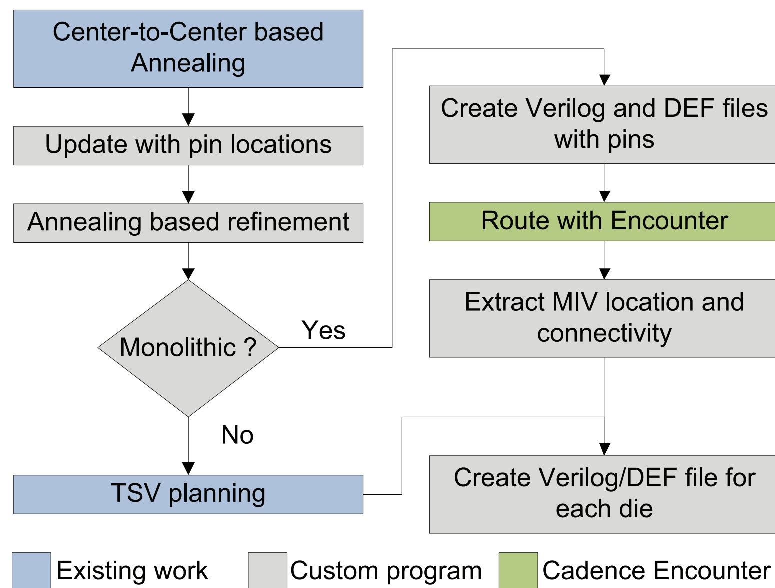


Figure 33: The design flow to obtain a 3D floorplan, assuming hard blocks.

3.1.2 Floorplanning Engine

This step takes the description of all the blocks as well as the connectivity information and generates an output floorplan that minimizes a certain cost function. This cost function is different for TSV-based and monolithic 3D ICs. A simulated annealing engine similar to [29] is used, which maintains a separate sequence pair for each die. The following different moves are performed during the annealing process: (1) change aspect ratio of a block (or rotate in case of hard blocks), (2) swap two blocks in either the positive sequence, negative sequence, or both, and (3) move or swap two blocks between a pair of dies/tiers.

In TSV-based 3D, the number of TSVs need to be limited as they each occupy significant silicon area. Hence, the TSV-based 3D cost function is given as follows

$$C_{TSV} = \alpha WL + \beta A + \gamma N_{TSV} \tag{24}$$

In the above equation, WL represents the inter-block wirelength, A represents the chip area, and N_{TSV} represents the number of TSVs. Since the MIV size is negligible in monolithic 3D, the floorplanner doesn't need to artificially control their count. The monolithic 3D cost function is given as follows

$$C_{MIV} = \alpha' WL + \beta' A \tag{25}$$

Now, in a given block-level netlist, not all the nets are timing critical. More effort should be spent minimizing the nets that are the most critical, at the expense of non-critical nets. A histogram of the longest path delays (LPD) through each inter-block net for a benchmark are shown in Figure 34.

From this figure, it is observed that this distribution follows something resembling a Gaussian curve for the nets with LPD greater than $0.35ns$. There are very few nets that are the most critical, and the most effort should be spent trying to minimize their length. Weighting each net by the LPD through it makes the floorplanner timing aware.

In case of soft blocks, the pin locations are determined after floorplanning, and measuring the wirelength from the center of the block is adequate. However, for hard blocks,

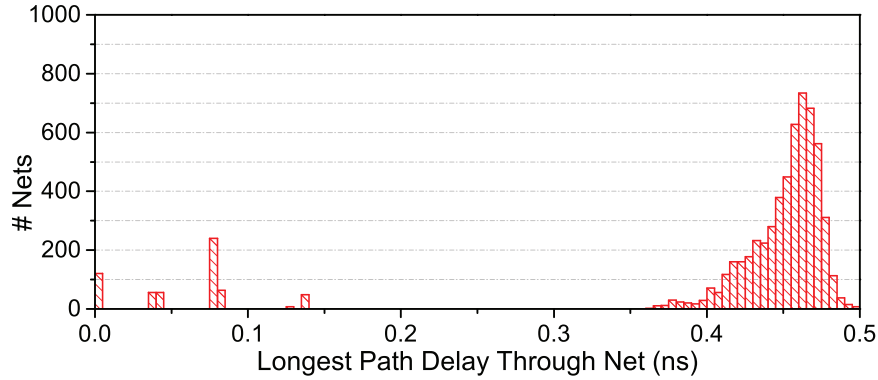


Figure 34: Histogram of the longest path delay through inter-block nets of a benchmark.

considering the pin locations of the blocks during floorplanning will require an extra step to compute the physical location of all block-pins. Since the number of block-pins are quite large, this will lead to large runtime overhead. Instead, a post-floorplanning refinement (PFPR) step is proposed to consider block-pin locations once the block locations have been determined.

3.1.3 Post-Floorplan Refinement (PFPR)

After determining the relative locations of all the blocks, each block is assumed to have a random orientation, and updated with its block-pin locations. Each block has 8 possible orientations, $0°$, $90°$, $180°$, $270°$, and their flipped counterparts. Without changing the relative locations of the blocks in the floorplan, each block can only have four possible orientations. For example, if the pins are in the center of a block, $0°$, $180°$ or $90°$, $270°$ and their flipped counterparts are all the same. However, if the pins are placed along the periphery each of the above four orientations gives a different wirelength result. The objective of this step is to determine the orientation of each block such that the wirelength is minimized. Simulated annealing is used for this purpose, where the only operation allowed is to change block orientation. The block orientation can only be changed among the allowed four scenarios. No sequence pair is necessary, as the relative locations of blocks do not change. Furthermore, wirelength computation can be done incrementally, as only one block is changed at a time.

3.1.4 MIV Planning Algorithm

Once the 3D floorplanning result is obtained, TSVs or MIVs need to be inserted into the whitespace between blocks. Since TSVs are big (around $5\mu m$ to $10\mu m$), and there may not be enough whitespace in the dies, a whitespace manipulation step is required. TSV planners exist, and this project uses the planner from [29] that constructs a 3D rectilinear Steiner tree (RST) from a 2D rectilinear Steiner minimum tree (RSMT). It then moves TSVs to nearby whitespace based on a network-flow formulation. If there is insufficient whitespace, it also inserts whitespace between blocks, at the cost of increased area.

However, in the case of monolithic 3D, MIVs are very small (around $70nm$), and it can be assumed that there is always whitespace available for MIV insertion. Since MIVs are also the same size as local vias, existing obstacle avoiding routers can be used to perform MIV insertion. Commercial tools, such as the 2D IC router in Cadence SOC Encounter can therefore be used. However, it is limited handling to 15 metal layers only. In order to maximize the number of dies that can be supported, three metal layers are used to represent the inter-block nets of a tier. For example, if a block is in tier 2, metal layer 4 is used to place block-pins, and metal layers 5 and 6 are used to represent inter-block routing on that tier. Vias between metal 6 and 7 represent MIVs between tier 2 and 3. The choice of the number of metal layers used is justified because only the inter-block nets are considered during MIV planning, and they are usually routed in the top 2 or 3 metal layers of each tier. Now, an MIV planning algorithm is presented assuming that the blocks are hard (block pin locations are known). Next, this is extended for soft-blocks.

3.1.4.1 MIV Planning for Hard Blocks

The MIV planning heuristic starts with creating a netlist that contains the connectivity information of the pins of all the 3D nets as shown in Lines 1–3 of Algorithm 2, where N_{net} denotes the total number of 3D nets. A DEF file that contains the physical location of every pin of each block is then created. $x^p_{b_i}$ and $y^p_{b_i}$ denote the x and y coordinates of

pin p of block b_i, respectively, and l^{b_i} denotes the metal layer that block b_i is assigned to. In addition, routing blockages for each block are added to account for: (1) the fact that MIVs cannot be placed within the blocks and (2) the internal wiring of each block (Lines 4–9). Next, verilog and DEF files are fed to SOC Encounter, which routes all the 3D nets simultaneously (Lines 10 and 11). Simultaneous routing of all 3D nets avoids any possible congestion issues due to the small size of MIVs. The routed DEF is parsed, and the routing topology of each net is traced to determine (1) the net that each MIV belongs to, and (2) the block-pin that each MIV connects to (Lines 12 and 13). Finally, verilog and DEF files for each tier (Lines 14 and 15) that contain the block/MIV locations are generated.

Algorithm 2: MIV planning algorithm for hard blocks.

```
   Input : Location of all blocks in B, block orientation, block-pin locations, and
           connectivity information
   Output: Number, location, and connectivity information of MIVs
 1 for n ← 1 to N_net do
 2 |   add connectivity information into a Verilog file;
 3 end
 4 for i ← 1 to |B| do
 5 |   for p ← 1 to N_pin^{b_i} do
 6 |   |   add pin physical location (x_{b_i}^p, y_{b_i}^p, l^{b_i}) in the DEF;
 7 |   end
 8 |   add routing blockage for b_i on its assigned layer l_j^{b_i};
 9 end
10 read the above Verilog and DEF files into SOC Encounter;
11 route the design and save the routed DEF file;
12 read the routed DEF file and reconstruct the routing graphs;
13 extract corresponding subnets in each die / tier from the routing graphs;
14 create Verilog file for each die/tier with subnet connectivity;
15 create DEF file for each die/tier with MIV locations;
```

3.1.4.2 MIV Planning for Soft Blocks

In the case of soft blocks, the block pin locations are determined only after floorplanning is finished. These block-pin locations are determined based on the inter-block connectivity, as well as the locations of any MIVs present. From the discussion on hard blocks, the MIV

locations depend on the block-pin locations as well. This is a chicken and an egg problem, and an iterative method to determine both the MIV and the block pin locations is presented in Figure 35.

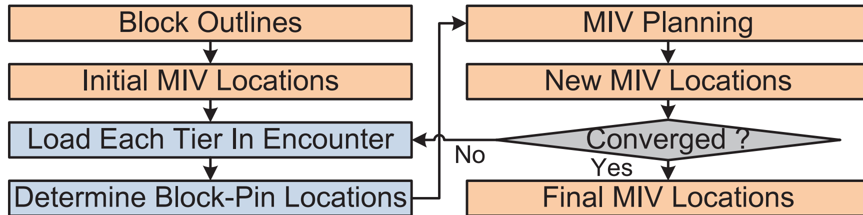


Figure 35: Iterative MIV planning algorithm for soft blocks.

Given the block outlines from the floorplanner, the blocks pins are first assumed to be in the center of the block. Next, for each 3D net, the optimal MIV location can then roughly be given as the center of its 3D bounding box. However, this approach will lead to overlap between blocks and MIVs, as well as between MIVs themselves, as shown in Figure 36(a).

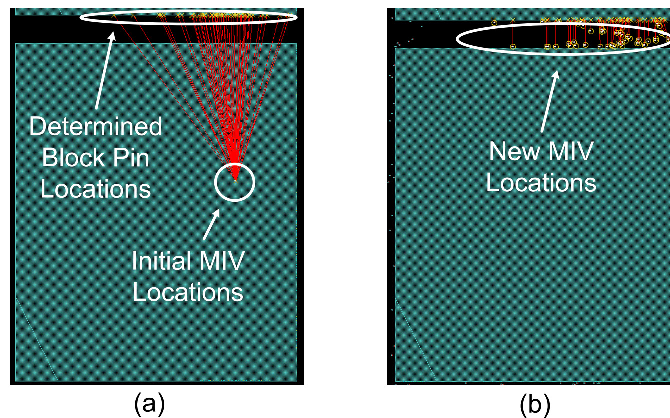


Figure 36: Illustration of MIV planning for soft blocks. (a) Initial estimated MIV locations (b) After one iteration of MIV planning.

With these initial MIV locations, verilog and DEF files are created for each tier. Cadence Encounter is then used to open each tier separately, and to determine the block pin locations based on the estimated MIV locations. These block pin locations can then be fed into the MIV planner for hard blocks to determine updated MIV locations, as shown in Figure 36(b). This entire process can be repeated until the MIV locations stabilize. In practice, only one or two iterations are required as the locations converge quickly. Once

the MIV locations are finalized, each block and tier can be placed and routed separately in Cadence Encounter.

3.2 Floorplan Quality Evaluation

This section evaluates the quality of the floorplan engine, as well as the quality of monolithic 3D vs TSV-based floorplans. All required code and scripts are implemented in C/C++ and python, and all experiments are carried out on a 2.5 GHz 64-bit linux system. The $45nm$ Nangate open source standard cell library is used in experiments. The TSV diameter, landing pad size, pitch, and thickness are assumed to be $6\mu m$, $7\mu m$, $10\mu m$, and $50\mu m$ respectively. The MIV diameter, pitch and thickness are $0.07\mu m$, $0.28\mu m$ and $0.31\mu m$ respectively. The TSV resistance and capacitance are $50m\Omega$, and $122fF$ respectively. These parasitics are measured values, taken from [64]. The MIV resistance and capacitance are similar to that of local vias and are 4Ω, and $1fF$ respectively, and six metal layers per tier are assumed.

3.2.1 Experimental Setup

Four benchmarks are considered, and their statistics are shown in Table 8. The first three are taken from the Opencores benchmark suite [51], and the fourth is a custom built 256-bit integer multiplier. This multiplier is built out of 256×4-bit multiplier and 512-bit adder blocks, arranged into an adder tree. Each multiplier block has 3 pipeline stages and each adder block has 4 pipeline stages.

Table 8: Design Statistics for All Benchmarks

Design	# Gates	#Blk	#Inter-blk nets	Intra-blk WL (μm)	Target period (ns)
des_perf	33,024	38	2,378	210,488	0.9
cf_rca_16	146,542	95	3,135	1,210,618	1.3
cf_fft_256_8	288,145	49	1,402	4,490,813	1.5
mult_256_256	1,639,050	127	49,471	12,354,340	0.845

In this particular section, evaluation is carried out with hard blocks, and the design flow used to obtain all results is shown in Figure 37. It consists of roughly two steps: block design, and top-level design and analysis.

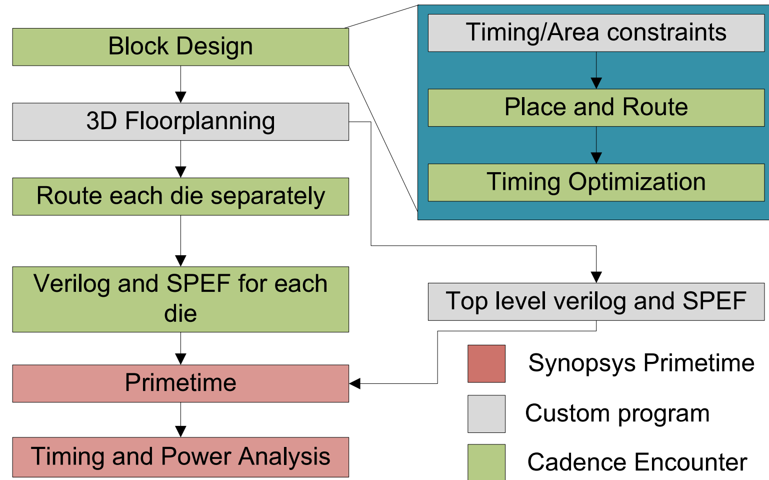


Figure 37: Our design flow used to get post-layout simulation results.

Each block is first designed separately in Cadence SoC Encounter. The netlist for each block is obtained by grouping modules bottom up along the hierarchy, until they reach a certain area threshold. Timing constraints for each block depend on the overall system frequency, and are determined by context characterization. Each block is then placed, routed and timing optimized in SOC Encounter. This step finalizes the pin locations within each block.

These blocks are then fed into the floorplanner to obtain block and MIV locations. After each die is routed separately in SOC Encounter, parasitic extraction is performed to obtain the SPEF files for each die. In addition, a top-level verilog file and SPEF file are created which contain inter-die connectivity and TSV/MIV parasitics, respectively. All netlist and parasitic information is then fed into Synopsys Primetime to obtain true 3D timing and power numbers. Sample layouts of block design as well as 2D floorplanning and 2-Die implementations of cf_rca_16 are shown in Figure 38.

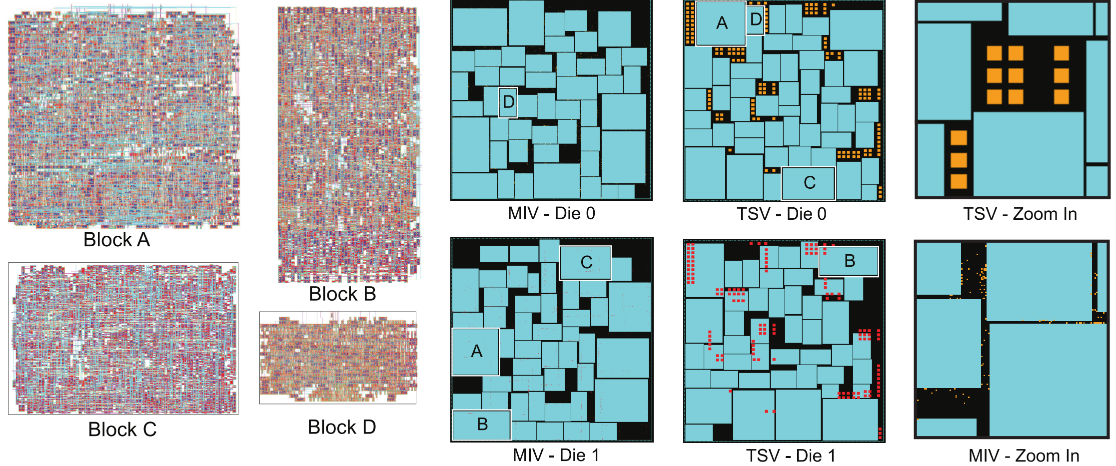


Figure 38: Sample layouts for cf_rca_16 testcase, along with select block designs, and zoomed in shots of TSVs and MIVs

3.2.2 Floorplanner Validation

The proposed floorplanner is run in 2D mode, and compared with the results obtained from wirelength-driven floorplanning in Cadence Encounter. The Encounter footprint area is obtained by gradually increasing the area and running floorplanning until no block overlap is observed. The results are summarized in Table 9.

Table 9: Comparison between the proposed floorplanner and Cadence Encounter.

Design	Footprint (mm^2)		Inter-block WL (m)	
	Encounter	This Project	Encounter	This Project
des_perf	0.0655 (1.00)	0.0604 (0.92)	0.352 (1.00)	0.356 (1.01)
cf_rca_16	0.445 (1.00)	0.413 (0.93)	0.361 (1.00)	0.368 (1.02)
cf_fft_256_8	1.690 (1.00)	1.141 (0.68)	0.414 (1.00)	0.437 (1.06)
mul_256_256	5.198 (1.00)	4.896 (0.94)	17.01 (1.00)	17.87 (1.05)
Average	1.00	0.87	1.00	1.035

As seen from this table, the proposed floorplanner produces comparable results with SOC Encounter. The large area reduction in the cf_fft_256_8 design is due to the fact that Cadence Encounter repeatedly produces module overlaps when provided with smaller area. This is presumably due to some bug in the legalization stage of SOC Encounter. It can still provide comparable wirelength to our floorplanner, however, as this particular testcase is only locally connected, and each block communicates with only one or two neighbours.

3.2.3 Monolithic 3D vs. TSV-based 3D

This section compares the intra-block as well as inter-block wirelength for each design implemented in 2D as well as monolithic or TSV-based 3D. These results are summarized in Table 10. From this table, it is observed that with respect to the inter-block wirelength, monolithic 3D gives significant advantage over 2D. The total wirelength reduction depends upon the ratio of inter-block wirelength to intra-block wirelength, and varies depending on the circuit. TSV-based 3D design however, does not give any improvement in wirelength for the small design des_perf, and small improvements begin be seen in the cf_rca_16 and cf_fft_256_8 testcases. However, with the largest design, no improvement is visible mainly because a large distance needs to be traversed from the block boundary to the nearest whitespace block to insert TSVs.

Therefore, monolithic 3D can provide significant benefits over 2D even in the case of small designs, while TSV-based 3D is suitable for designs with a large number of long interconnections or memory-on-logic stacking applications.

3.3 Inter-Tier Performance Differences

Although it has been demonstrated that monolithic 3D ICs offer significant advantages, it has so far been assumed that both tiers have identical performance. In reality, one or more of the tiers suffers from degraded performance, due to limitations in the current fabrication process. This section discusses the source of these differences and how to model them.

3.3.1 Source of Inter-Tier Performance Differences

The fabrication process was shown in Figure 1. During the fabrication process of the top tier, a low temperature transistor process is key to prevent damage to the devices of the bottom tier. It has been demonstrated [65] that transistors can be fabricated at temperatures down to $625°C$ without any loss of performance. While this is sufficient to prevent damage to the devices, this temperature is still too high to prevent damage to the copper BEOL.

Table 10: A comparison of wirelength, timing and top net power of 2D versus 3D

Type		Footprint ($\mu m \times \mu m$)	Norm. Si. Area	#MIV/ #TSV	Inter-block routed WL (μm)	Total routed WL (μm)
des_perf						
2D	Encounter	256x256	1	-	352,805 (1.00)	563,293 (1.00)
	Ours	251x241	0.92	-	356,489 (1.01)	566,977 (1.01)
MIV	2 Tiers	146x211	0.94	1,800	267,678 (0.76)	478,166 (0.85)
	3 Tiers	127x179	1.04	2,738	222,240 (0.63)	432,728 (0.77)
	4 Tiers	111x149	1.01	3,823	204,868 (0.58)	415,356 (0.74)
TSV	2 Dies	215x323	2.12	120	473,092 (1.34)	683,580 (1.21)
	3 Dies	320x235	3.44	456	515,267 (1.46)	725,755 (1.29)
	4 Dies	359x402	8.81	984	734,739 (2.08)	945,227 (1.68)
cf_rca_16						
2D	Encounter	667x667	1	-	361,673 (1.00)	1,572,291 (1.00)
	Ours	555x744	0.93	-	367,542 (1.02)	1,578,160 (1.00)
MIV	2 Tiers	416x477	0.89	1,747	289,156 (0.80)	1,499,774 (0.95)
	3 Tiers	367x370	0.92	2,925	255,910 (0.71)	1,466,258 (0.93)
	4 Tiers	273x384	0.94	3,936	240,583 (0.67)	1,451,201 (0.92)
TSV	2 Dies	484x418	0.91	156	354,347 (1.07)	1,564,965 (1.00)
	3 Dies	377x370	0.94	334	401,425 (1.11)	1,612,043 (1.03)
	4 Dies	350x349	1.10	477	345,090 (0.95)	1,555,708 (0.99)
cf_fft_256_8						
2D	Encounter	1,300x1,300	1.00	-	413,674 (1.00)	4,904,487 (1.00)
	Ours	1,142x999	0.68	-	436,933 (1.06)	4,927,746 (1.00)
MIV	2 Tiers	819x718	0.70	1,050	263,787 (0.64)	4,754,600 (0.97)
	3 Tiers	581x799	0.82	1,921	254,256 (0.61)	4,745,069 (0.97)
	4 Tiers	595x594	0.84	2,475	269,049 (0.65)	4,759,862 (0.97)
TSV	2 Dies	679x932	0.75	75	369,166 (0.89)	4,859,979 (0.99)
	3 Dies	653x674	0.78	147	357,592 (0.86)	4,848,405 (0.99)
	4 Dies	584x527	0.73	377	422,216 (1.02)	4,913,029 (1.00)
mult_256_256						
2D	Encounter	2,280x2,280	1.00	-	17,089,968 (1.00)	29,444,308 (1.00)
	Ours	2,144x2,284	0.94	-	17,870,346 (1.05)	30,224,686 (1.03)
MIV	2 Tiers	1,506x1,718	1.00	48,513	13,815,376 (0.81)	26,169,716 (0.89)
	3 Tiers	1,286x1,295	0.96	79,682	11,392,196 (0.67)	23,746,536 (0.81)
	4 Tiers	1,177x1,131	1.02	102,994	10,116,222 (0.59)	22,470,562 (0.76)
TSV	2 Dies	1,608x1,616	1.00	1,683	18,825,744 (1.10)	31,180,084 (1.06)
	3 Dies	1,508x1,236	1.08	3,599	21,184,404 (1.24)	33,538,744 (1.14)
	4 Dies	1,240x1,190	1.14	4,232	20,890,062 (1.22)	33,244,402 (1.13)

This problem can be avoided by using tungsten as the interconnect material on the bottom tier [2], which degrades the interconnects. If, however, copper must be used in the bottom tier, the top tier needs an alternate manufacturing process such as laser scan anneal [55], which degrades the top-tier transistors. Therefore, the choice is between degraded interconnects on the bottom tier or degraded transistors on the top tier. This section discusses the modelling of these performance degradations.

3.3.2 Degraded Interconnects on the Bottom Tier

Tungsten has several attractive properties that make it a suitable choice for nano-scale interconnects. It has a much higher melting point than copper ($3422°C$ vs $1085°C$), so no low temperature process is needed for the top tier. It also does not diffuse into silicon, eliminating the need for a diffusion barrier and preventing any copper contamination issues during FEOL processing of the top tier. It also has much higher electromigration resistance and can be etched similar to aluminium, eliminating the need for a damascene process. However, tungsten has a bulk resistivity $3.1\times$ that of copper, which has so far prevented its widespread use.

When interconnects shrink, the bulk resistivity no longer applies, and resistivity goes up due to effects such as line edge roughness, sidewall scattering, and grain boundary scattering. The equation for size dependent resistivity of an interconnect is given by [42]:

$$\rho_{eff} = \frac{\rho_0}{\sqrt{1-(u/w_0)^2}} \left\{ \left[\frac{1}{9} - \frac{\alpha}{6} + \frac{\alpha^2}{3} - \frac{\alpha^3}{3} ln\left(1+\frac{1}{\alpha}\right) \right]^{-1} + 0.45(1-p)\ \frac{\lambda}{w_0}\left(\frac{w_0}{h_0} + \frac{1}{1-(u/w_0)^2}\right) \right\} \tag{26}$$

Most of these quantities are empirically fitted, and an explanation of the various parameters and a choice of their values for both copper and tungsten are listed in Table 11.

Using this equation, the resistivity for both tungsten and copper interconnects are plotted in Figure 39. This curve is in close agreement with measured data from IBM [6]. It is observed that the degradation of resistivity due to tungsten is significantly lower at lower

Table 11: Various interconnect parameters

Parameter	Description	Copper	Tungsten
w_0	Width		
ρ_0	Bulk Resistivity ($\mu\Omega$-cm)	1.68	5.28
u	Line Edge Roughness	$0.4w_0$	$0.4w_0$
h_0	Height (Thickness)	$1.8w_0$	$1.8w_0$
d	Dist. Between Grain Boundaries	w_0	w_0
λ	Electron Mean Free Path (nm)	39 [54]	19.1 [9]
p	Sidewall Specularity	0.2 [54]	0.3 [60]
R	Grain Boundary Reflectivity	0.3 [54]	0.25 [60]
α	$\lambda R/(dR(1-R))$		

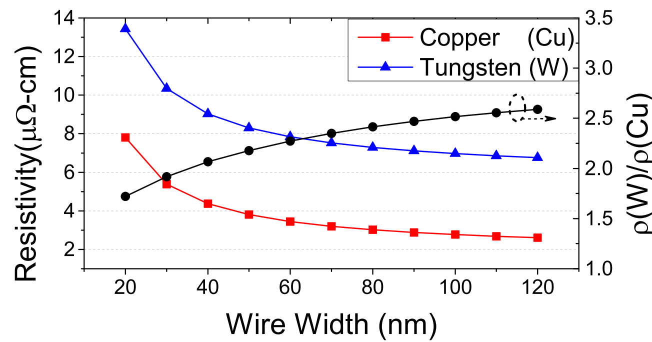


Figure 39: Copper vs. Tungsten resistivity at different wire widths.

widths. It should be noted that a $3nm$ diffusion barrier was assumed for both tungsten and copper. In reality, tungsten does not diffuse into the ILD, and a diffusion barrier is not strictly necessary. This makes the tungsten numbers pessimistic, and its resistivity will be lower in practice.

Using these resistivity values, the change in the interconnect resistivity for the Nangate 45nm library is tabulated in Table 12. From this table, it is observed that the local metal lines degrade less than the global metal lines. These modified resistivity values are used to generate interconnect technology file (.ict), and fed into Cadence QRC Techgen to re-characterize the interconnect extraction libraries.

3.3.3 Degraded Transistors on the Top Tier

If copper is to be used on the bottom tier, laser-scan anneal has been proposed for the dopant activation on the top tier [55]. This results in localized heating in the source/drain

Table 12: The change in resistivity values of different metal layers in the Nangate 45nm library due to Tungsten interconnects.

Layer	Width(nm)	Thickness(nm)	ρ(Cu) / ρ(W)
Metal1 – Metal3	70	140	2.38
Metal4 – Metal6	140	280	2.67
Metal7 – Metal8	400	800	2.94
Metal9 – Metal10	800	2000	3.04

regions thereby preventing damage to the underlying devices and interconnects. However, the process is not mature yet, and identical transistor performance as a high-temperature anneal has not yet been obtained. The PMOS and NMOS performance degrade by 27.8% and 16.2% respectively [55]. This is referred to as the $TTm20p$ corner, as on average, the performance is worse by roughly 20%. However, this work was from several years ago, and improvements in the process are bound to be made. Therefore, to represent fabrication progress, another corner $TTm10p$ is defined, which has a PMOS and NMOS degradation of 13.9% and 8.1% respectively, which is exactly half that of the $TTm20p$ corner. The transistor parameters in the Nangate 45nm library are modified to represent this degradation, and the IV curves of the nominal and degraded transistors are plotted in Figure 40.

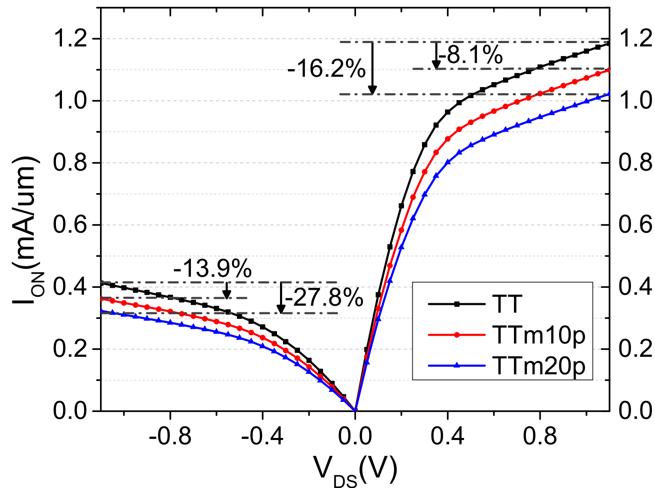


Figure 40: IV curves of nominal and degraded transistors.

These modified transistor models are used to re-characterize all the std. cell libraries using Encounter Library Characterizer. The resulting performance of select std. cells at

maximum loading is tabulated in Table 13. In addition to re-characterization at different transistor corners, tungsten interconnects also increase the internal parasitics of std. cells. The std. cells are also re-characterized under this condition, and this corner is named TT_W.

Table 13: Minimum size (X1) std. cell average delay (in ps), assuming worst loading, at different corners.

Std. Cell	TT	TTm10p	TTm20p	TT_W
NAND2	221.8 (1.00)	243.9 (1.10)	265.2 (1.19)	222.35 (1.00)
AOI211	154.5 (1.00)	173.8 (1.12)	192.9 (1.25)	154.97 (1.00)
XOR2	163.42 (1.00)	187.6 (1.14)	210.85 (1.28)	163.86 (1.00)
DFF Clk-Q	213.1 (1.00)	243.8 (1.14)	277.7 (1.30)	214.05 (1.00)
DFF Setup	40.29 (1.00)	50.95 (1.26)	58.11 (1.44)	43.86 (1.08)

From this table, it is observed that the cell delays for simple gates such as NAND roughly follow the average of NMOS and PMOS degradation, while complex gates are more or less dominated by PMOS degradation. In addition, it is observed that the setup time for the flip-flops degrade at a much higher rate than either NMOS/PMOS. Tungsten interconnects only have a minimum impact on the gate performance, as the wires within the std. cells are very small, and the resistance is dominated by the R_{ON} of the transistor.

In summary, two choices exist: (1) Use tungsten on the bottom tier and deal with degraded interconnects and marginally worse std. cells, or (2) Use copper on the bottom tier and deal with significantly degraded std. cells on the top tier. This chapter studies both options and compares and contrasts them.

3.4 Performance-Difference-Aware Design and Analysis Flow

This section first describes how the floorplanner is modified such that designs become less sensitive to inter-tier performance differences. It then describes how timing and power analysis is performed for a 3D design where each tier has different models for transistors or interconnects.

3.4.1 Performance-Difference-Aware Floorplanner

In most designs, not every block is timing critical. Although non-timing critical blocks can operate faster, they are synthesized at the frequency of the critical block to save area and power. Therefore, even with degraded transistors, these blocks can be synthesized to operate at the frequency of the critical block, albeit with a larger area. As long as the critical blocks do not operate with slower transistors or interconnects, the chip can still meet timing.

Given the block RTL and timing constraints, four different versions of each block are synthesized: One for the nominal corner, and one for each of the degraded libraries. In the case of tungsten interconnects, in addition to the modified standard cell libraries, the resistivity of the wire load models is modified to accurately drive synthesis. For each version of the block, the area and longest path delay (LPD) through it are noted. An illustration of this synthesis is shown in Figure 41, where all the blocks in a particular design are synthesized at all four corners.

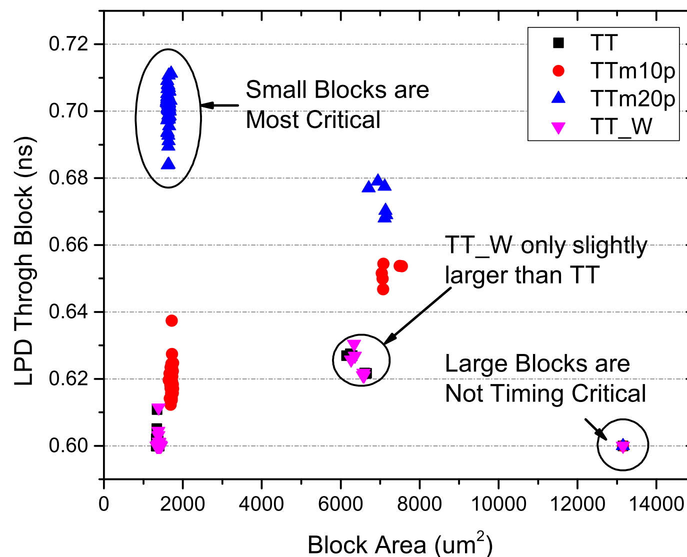


Figure 41: Synthesis results of “des3” benchmark for different degradations.

This figure plots the block area vs. the longest path delay through it. Each point on this plot is a single block. As seen from this graph, the largest blocks in this benchmark are not

timing critical. For all of the degraded transistor and interconnect options, they have the same frequency and area. However, the smallest blocks seem to be the most timing critical. They require much larger area (buffers) to try and meet timing, and it is still not possible.

Given that the design has inter-tier performance differences, each block will have a different area and LPD depending on the tier in which it lies. The premise of the modified floorplanner is to move the timing critical blocks to the tier that is not degraded. The non-timing critical blocks, although on a slower tier, can be optimized to meet the system frequency. An overview of the inter-tier performance-difference aware floorplanner is shown in Figure 42.

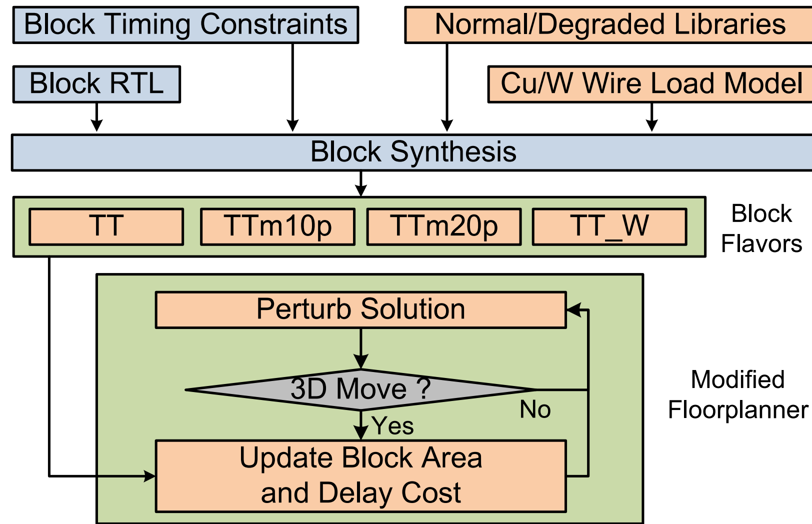


Figure 42: The proposed inter-tier performance difference aware floorplanner.

If LPD(b_i) is the tier-dependant longest path delay of a block b_i, the modified cost function of the floorplanner is defined as:

$$Cost_{VA} = \alpha.WL + \beta.Area + \gamma \sum_{i=1}^{N_{Block}} LPD(b_i) \tag{27}$$

In the above equation, WL refers to the wirelength. The area of a block is also dependent on its tier. Therefore, whenever a 3D move is made, the area of all the blocks that have changed their tier is also updated. The third term in the above equation will try to place the timing critical blocks in the faster tier, and the non-timing critical blocks in the slower tier.

An illustration of the modified floorplanner is shown in Figure 43. This figure assumes that the top tier is at the $TTm20p$ corner, and the floorplanning is carried out for the same benchmark shown in Figure 41. In this figure, it is observed that the performance difference aware floorplanner moves the smaller, more timing critical blocks to the bottom tier, so they can operate at the desired frequency.

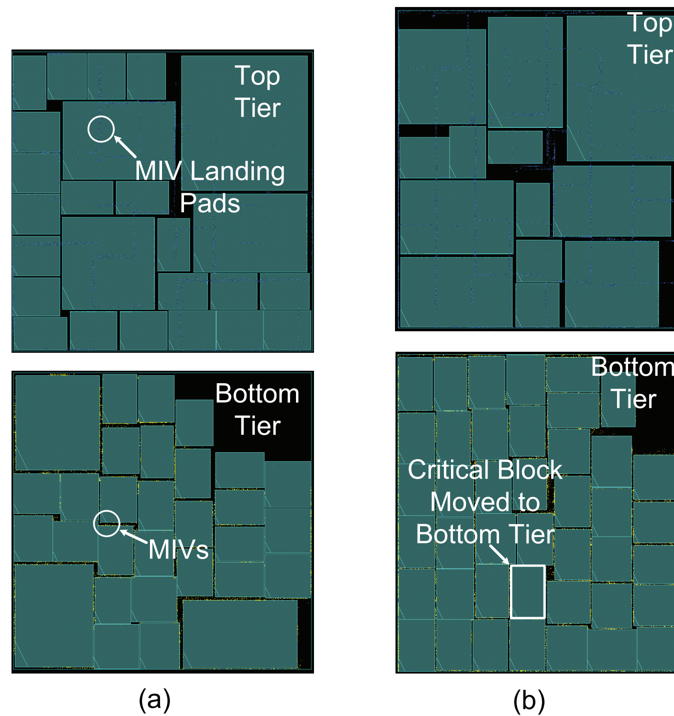


Figure 43: Floorplan screenshots of “des3” when the top tier is at the TTm20p corner. (a) Without performance difference aware floorplanning, and (b) With performance difference aware floorplanning.

3.4.2 Performance-Difference-Aware Analysis

The floorplanner gives the corner in which each block operates. Once the placed and routed netlists of all the blocks and tiers are available, they are loaded into Synopsys PrimeTime. The appropriate std. cell library is chosen for each cell depending on the tier in which it lies. The extraction tech file for each block and tier is also modified depending on the interconnect material, and the appropriate parasitics are loaded into Synopsys PrimeTime.

A top-level netlist and parasitic file is created to represent the MIV connectivity and parasitics. According to [2], if the inter-tier oxide thickness is greater than or equal to $100nm$, there is negligible inter-tier coupling. Therefore, any such coupling is ignored. Once all the netlists and parasitics are loaded, 3D timing and statistical power analysis is performed.

3.5 Power-Performance Study

One benchmark is chosen from the OpenCores benchmark suite (des3), one from the IWLS benchmark suite (b19), and one custom 128-bit integer multiplier is designed. These benchmarks are designed using the Nangate 45nm library, and their statistics are tabulated in Table 14. The cell counts shown are the synthesis results without any wire load models. In all the 3D implementations considered, the diameter of an MIV is assumed to be $100nm$, with a resistance of 2Ω and a capacitance of $0.1fF$ [34].

Table 14: Benchmarks used for evaluation evaluation.

Benchmark	#Blocks	#Gates	#Inter-Block Nets
des3	55	63,194	6,138
b19	55	78,852	14,223
mul128	63	253,867	12,447

3.5.1 Identical Performance on Both Tiers

This section discusses the case where both tiers in 3D have identical transistors and interconnects. This represents an ideal manufacturing process, and represents the best possible case for monolithic 3D. Initial floorplanning is first performed to derive wire load models for each benchmark. Floorplanning is carried out again, and basic floorplan comparisons for 2D and 3D are tabulated in Table 15. In addition to these two flavors, an "ideal" block-level implementation is defined. This implementation is obtained by assuming that all the inter-block nets have zero length and parasitics. During the block implementation, the output load of the blocks is set to be zero and the inputs are assumed to be driven by ideal drivers. This is the lower bound on any block-level implementation of this design, given the same set of blocks, and the constraint that each block is implemented in 2D.

Table 15: Basic floorplan comparisons assuming both tiers have same performance.

Ckt.	Flavor	#Gates ($\times 10^3$)	Footprint (mm^2)	Total WL (m)	# MIV ($\times 10^3$)
des3	2D	68.9 (1.00)	0.328 (1.00)	1.514 (1.00)	-
	3D	66.2 (0.96)	0.156 (0.48)	1.287 (0.85)	3.75
	Ideal	64.4 (0.94)	-	0.938 (0.62)	-
b19	2D	82.3 (1.00)	0.398 (1.00)	3.341 (1.00)	-
	3D	80.62 (0.98)	0.204 (0.51)	2.847 (0.85)	13.46
	Ideal	79.35 (0.96)	-	1.838 (0.55)	-
mul128	2D	251 (1.00)	1.096 (1.00)	4.693 (1.00)	-
	3D	245 (0.97)	0.550 (0.50)	4.447 (0.95)	7.261
	Ideal	235 (0.93)	-	3.271 (0.70)	-

From this table, it is observed that monolithic 3D leads to significantly shorter wirelength. Although the inter-block wirelength is always significantly reduced, the total wirelength reduction depends on the intra-block wirelength as well. Benchmarks such as "mul128" have most of the wirelength within the block, so there is not much total wirelength reduction. In addition, shorter wires leads to fewer gates (buffers) being required.

Next, the power-performance trade-off for each of the three different implementations is studied. In order to get the numbers for the ideal implementation, the parasitics of all inter-block nets are forced to zero in Synopsys PrimeTime. In addition to the nominal V_{DD} of $1.1V$, the std. cell libraries are characterized at four additional V_{DD} values covering $\pm 10\%$ of nominal V_{DD} (1.00V, 1.05V, 1.10V, 1.15V, 1.20V). The power and frequency are measured at each of these V_{DD} values and the resulting curves are plotted in Figure 44.

From this figure, it is observed that 3D usually offers a performance advantage (at the same power) over 2D, and it closes the gap to ideal by up to 50%. This additional performance can be traded for power savings to meet the 2D frequency, and up to a 16.1% reduction in power is observed. In these curves, the ideal implementation of "b19" requires extrapolation to make iso-performance power comparisons at the nominal 2D frequency. Such a comparison is avoided due to inaccuracies that are bound to be introduced by extrapolation.

The reason the absolute values of the gains in the "mul128" benchmark are so small

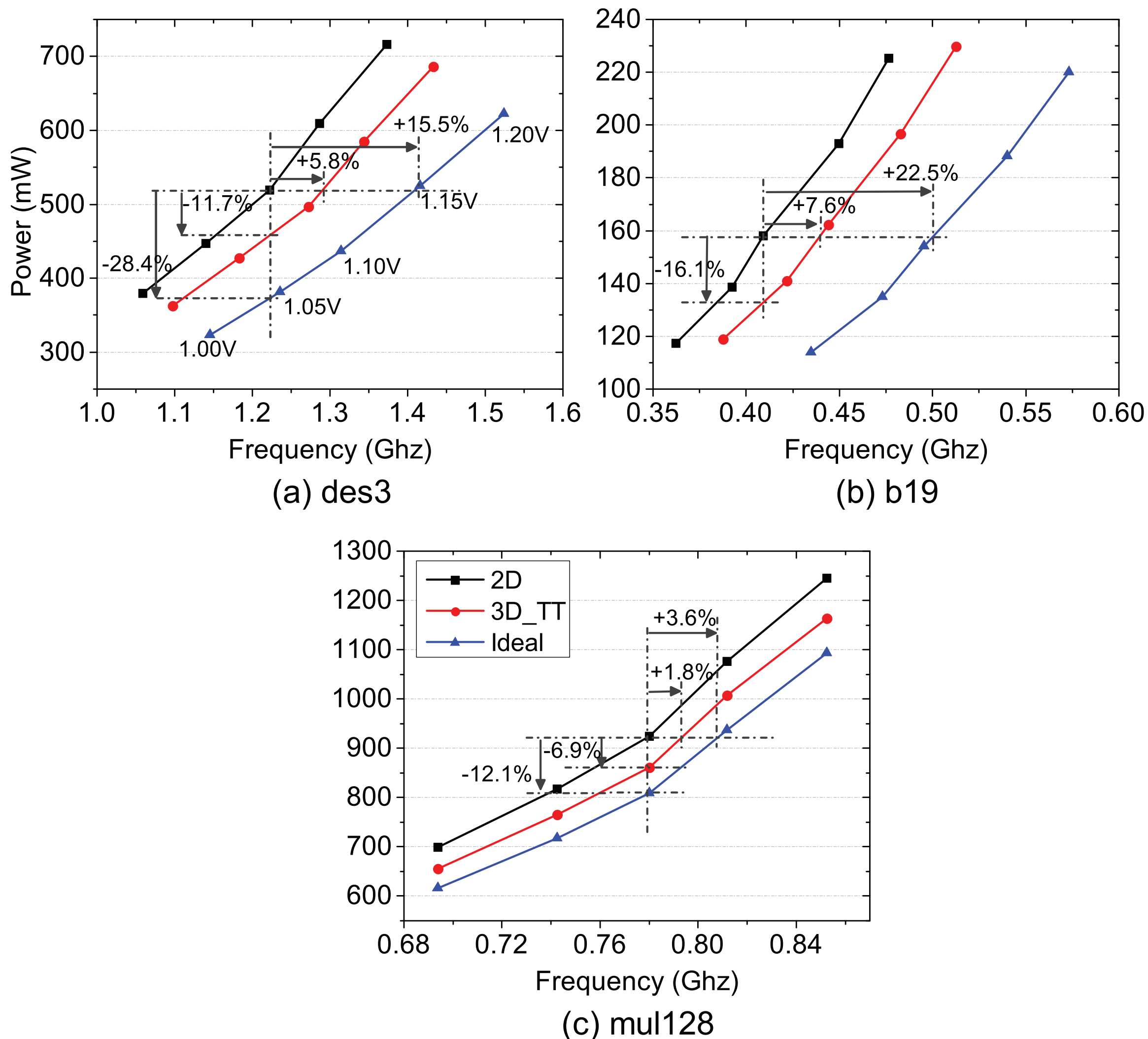


Figure 44: Power-performance trade-off curves assuming that both the tiers have identical transistors and interconnects.

is because the critical path is always within a single block. Since the inter-block nets are not timing critical, shortening them does not make the design faster, and there is no additional performance to trade for power. Making this design faster will require architectural modifications such as block folding.

3.5.2 Impact of Inter-Tier Performance Differences

The performance difference aware floorplanner (PDAFP) is run on all benchmarks for each degraded option, and the basic floorplan comparisons are tabulated in Table 16. The numbers are normalized to the respective 2D numbers in Table 15.

Table 16: Basic floorplan comparisons for different degraded 3D options. The numbers are normalized to the respective 2D numbers in Table 15.

Ckt.	Flavor	#Gates ($\times 10^3$)	Footprint (mm^2)	Total WL (m)	# MIV ($\times 10^3$)
des3	Top=TTm10p	68.1 (0.99)	0.159 (0.49)	1.29 (0.85)	3.92
	Top=TTm20p	67.2 (0.98)	0.177 (0.54)	1.44 (0.95)	5.67
	Bot=TT_W	66.8 (0.97)	0.153 (0.47)	1.31 (0.87)	3.11
b19	Top=TTm10p	80.8 (0.98)	0.212 (0.53)	2.84 (0.85)	11.6
	Top=TTm20p	82.0 (1.00)	0.222 (0.56)	2.90 (0.87)	11.3
	Bot=TT_W	80.8 (0.98)	0.208 (0.52)	2.91 (0.87)	12.9
mul128	Top=TTm10p	247 (0.98)	0.574 (0.52)	4.35 (0.93)	4.48
	Top=TTm20p	249 (0.99)	0.575 (0.52)	4.38 (0.94)	4.48
	Bot=TT_W	246 (0.98)	0.568 (0.52)	4.29 (0.91)	4.48

As observed from this table, all of the degraded options use more gates than the case when both tiers have identical performance. However, the gate counts are still less than 2D. Similarly, both the footprint area and the wirelength are increased from the non-degraded case, but are still less than 2D. The only exception is the "mul128" benchmark, when the bottom tier is at the TT_W corner. This has a slightly lower wirelength than the non-degraded option, but this is due to the trade off with footprint area.

Next, the power-performance trade-off curves for the degraded transistors and interconnects are plotted in Figure 45. For the sake of comparison, degraded transistors and interconnects on top of a non PDAFP floorplanning solution are also plotted.

As observed from this figure, the performance difference aware floorplanner (PDAFP) always outperforms the non-PDAFP one. The top tier having TTm20p transistors is always worse than 2D, except in the case of "mul128". After PDAFP, the top tier with TTm10p transistors always becomes better than 2D. Finally, tungsten interconnects on the bottom tier are by far the best option, and although there is negligible timing degradation compared to the identical tiers case, some power overhead exists.

To summarize the impact of PDAFP, the iso-power frequency and iso-performance power for differetn benchmarks are tabulated in Table 17. The comparison point for each of

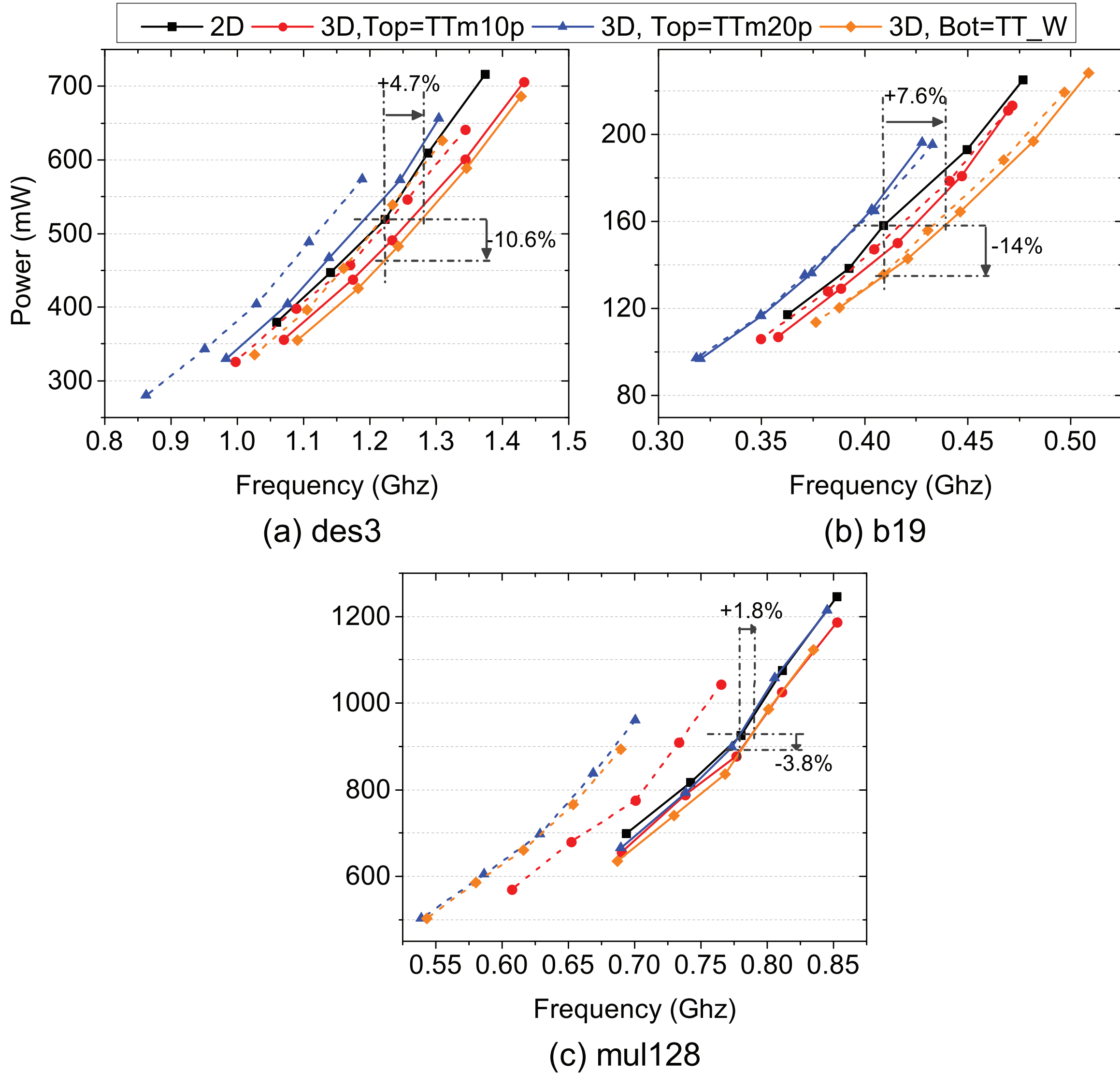


Figure 45: Power-performance trade-off curves assuming degraded transistors and interconnects. Dashed lines represent non performance difference aware floorplanning and solid lines represent performance difference aware floorplanning.

the three benchmarks is the respective 2D power and frequency at nominal V_{DD}. If a particular point is not achievable within $\pm 10\%$ of nominal V_{DD}, and extrapolation is required, it is marked with a '-'.

From this table, PDAFP improves the iso-power performance by up to 12.6% and the iso-performance power by up to 10.6%. The non-PDAFP floorplan results are often not able to meet the 2D frequency even with a 10% V_{DD} boost. If the V_{DD} was increased further so that they could meet timing, PDAFP would show even more benefit.

Table 17: Impact of performance difference aware floorplanning (PDAFP). '-' indicates that point is not achievable within $\pm 10\%\ V_{DD}$.

Ckt.	Parameter	Top=TTm10p		Top=TTm20p		Bot=TT_W	
		Non-PDAFP	PDAFP	Non-PDAFP	PDAFP	Non-PDAFP	PDAFP
des3	iso-power freq. (Ghz)	1.233	1.259 (+2.1%)	1.14	1.19 (+4.4%)	1.222	1.28 (+4.7%)
	iso-freq. power (mW)	507.746	479.1 (-5.6%)	-	547.65 (-)	519.48	464.55 (-11.6%)
b19	iso-power freq. (Ghz)	0.417	0.424 (+1.7%)	0.396	0.396 (+0%)	0.432	0.439 (+1.6%)
	iso-freq. power (mW)	151.723	144.58 (-4.7%)	173.14	172.828 (-0.2%)	135.06	135.06 (+0%)
mul128	iso-power freq. (Ghz)	0.737	0.793 (+7.6%)	0.692	0.779 (+12.6%)	-	0.793 (-)
	iso-freq. power (mW)	-	892.95 (-)	-	922.53 (-)	-	887.37 (-)

Table 18: Iso-power performance and iso-performance power results for all implementation flavors.

Ckt.	Parameter	2D	Ideal	3D			
				Both=TT	Top=TTm10p	Top=TTm20p	Bot=TT_W
des3	iso-power freq. (Ghz)	1.222	1.411 (+15.5%)	1.293 (+5.8%)	1.259 (+3.0%)	1.19 (-2.6%)	1.28 (+4.7%)
	iso-freq. power (mW)	519.48	372.06 (-28.4%)	458.45 (-11.7%)	479.1 (-7.8%)	547.65 (+5.4%)	464.55 (-10.6%)
b19	iso-power freq. (Ghz)	0.408	0.5 (+22.5%)	0.439 (+7.6%)	0.424 (+3.9%)	0.396 (-2.9%)	0.439 (+7.6%)
	iso-freq. power (mW)	157.05	- (-)	131.81 (-16.1%)	144.58 (-7.9%)	172.828 (+10.0%)	135.06 (-14.0%)
mul128	iso-power freq. (Ghz)	0.779	0.807 (+3.6%)	0.793 (+1.8%)	0.793 (+1.8%)	0.779	0.793 (+1.8%)
	iso-freq. power (mW)	922.53	810.56 (-12.1%)	859.15 (-6.9%)	892.95 (-3.2%)	922.53	887.37 (-3.8%)

3.5.3 Overall Comparisons

The iso-power performance and iso-performance power for 2D, ideal, the non-degraded monolithic 3D, as well the PDAFP results for degraded monolithic 3D are tabulated in Table 18.

From this table, it is clearly seen that tungsten interconnects on the bottom tier outperform degraded transistors on the top tier. This option is preferable from the manufacturing perspective as well, as the process is already available. Even with tungsten interconnects on the bottom tier, the gap to the ideal block-level implementation can be closed by up to 50% w.r.t. performance and 36% w.r.t power.

3.5.4 Block Folding

As mentioned in Subsection 3.5.1, the "mul128" benchmark has very limited benefit in block-level 3D due to the fact that the critical path is within a single block. This block is a 128×4 multiplier. In this benchmark, there are 32 such blocks. Each of these blocks has only 4,906 gates when synthesized without any wire load models, and is too small to be folded using other 3D technologies such as TSV-based 3D. This section demonstrates how monolithic 3D can help to increase the chip performance and decrease the chip power by folding this one block.

In order to perform 3D block folding, the gate-level 3D placer presented in [28] is used. Once the locations of all gates are determined, MIV insertion is performed by tricking the 2D router, similar to the method presented for block-level designs.

This block is first synthesized without any wire load models, implemented it in 2D and 3D, and then re-synthesized using the derived wire load models. This is then placed, and the resulting footprint and wirelength comparisons are shown in Table 19. The corresponding screenshots are shown in Figure 46.

From this table, block folding offers 26% wirelength reduction, even for extremely small blocks. The MIV density is approximately $50,000$ per mm^2, which is significantly

Table 19: Placement results for the 128×4 multiplier block.

Flavor	#Gates	Footprint (um^2)	WL (um)	# MIV
2D	5,398 (1.00)	13,225 (1.00)	61,045 (1.00)	-
3D	5,261 (0.97)	6,561 (0.50)	45,336 (0.74)	326

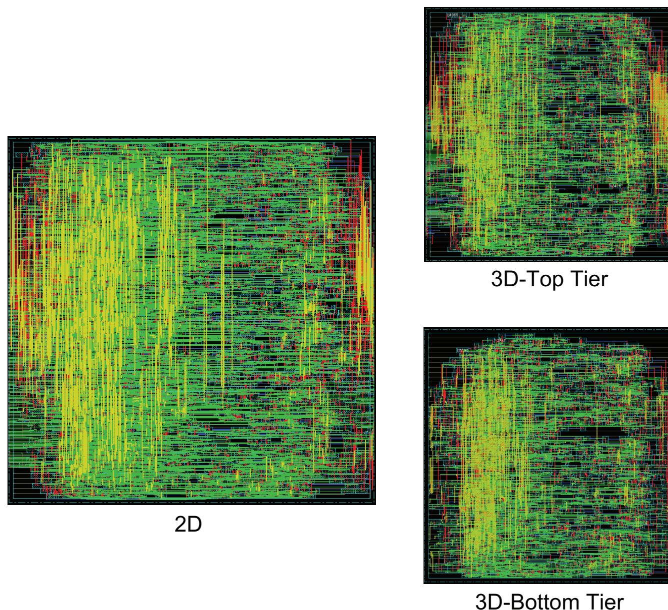


Figure 46: 3D placement layout snapshots of one 128×4 multiplier block within the "mul128" benchmark.

higher than that offered by any other 3D integration technology. In addition, this comes at zero area overhead.

Finally, similar to the block-level designs, the power-performance curves for 2D and 3D designs are plotted. In addition, since it has already been demonstrated that tungsten interconnects are preferable to degraded transistors, the power-performance curves are also plotted assuming that the bottom tier uses tungsten interconnects. These curves are shown in Figure 47.

As seen from this figure, even with degraded interconnects, a 5.7% performance boost and 12.6% power saving is obtained. The impact due to tungsten is minimal, as such small blocks are almost always transistor dominated. The above results suggest an alternate design methodology for monolithic 3D ICs. Every block is folded using tungsten interconnects on the bottom tier. This comes at a negligible performance hit, as the blocks

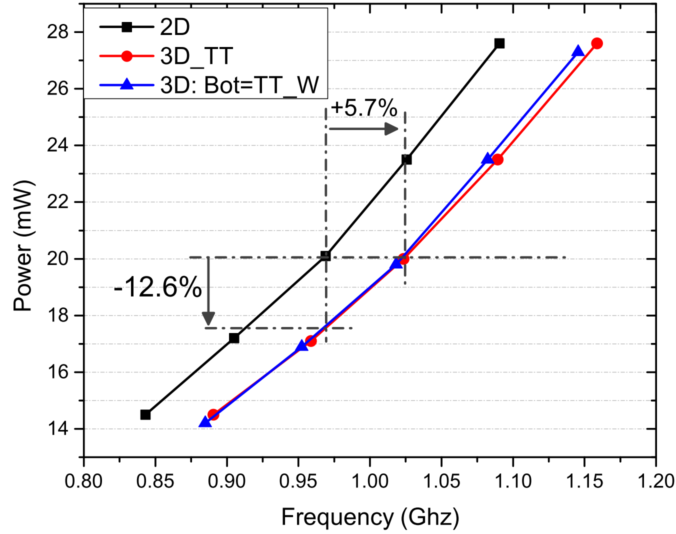


Figure 47: Power-performance trade-off curves for the 128×4 multiplier block.

are gate dominated. Next, since each block has a reduced footprint, assembling these 3D blocks together will reduce the chip footprint, leading to shorter wires between blocks. The timing critical buses between the blocks can then be routed using the global metal layers of the top tier, using copper interconnects, at no performance loss.

3.6 Summary

This chapter presented physical design techniques for block-level monolithic 3D ICs under real world considerations. First, a floorplanning framework was presented, and it was demonstrated that this engine produces results comparable to commercial engines. Next, it was demonstrated that even in coarse-grained integration such as block level, monolithic 3D significantly outperforms other 3D styles such as TSV-based 3D.

Inter-tier performance differences that arise due to an immature fabrication process was discussed, and two options for monolithic 3D ICs were discussed and modeled. A performance difference aware floorplanner was presented, and it was demonstrated that using this floorplanner, monolithic 3D still shows significant benefits compared to 2D ICs. Finally, it was demonstrated that tungsten interconnects on the bottom tier are preferable to degraded transistors on the top tier.

CHAPTER IV

PHYSICAL DESIGN FOR GATE-LEVEL MONOLITHIC 3D ICS

So far, block-level monolithic 3D ICs have been discussed. However, the potential benefit offered is limited, as this style does not fully take advantage of the high integration density offered. In contrast, the gate-level design style naturally lends itself to monolithic 3D ICs. Existing standard cells and memory can simply be reused, placed onto multiple tiers, and MIVs used to connect them together. In addition, there is no silicon area overhead of doing this. Out of the three design styles available for monolithic 3D ICs, gate-level offers the greatest balance between integration density and reuse of existing libraries. The authors of [4] provided a rudimentary design flow that is not capable of handling any hard macros such as memory, and therefore cannot be applied to real designs.

The gate-level design style can also be applied to other stacking technologies such as TSV-based 3D ICs and face-to-face 3D ICs. In TSV-based 3D ICs, the via size is so large compared to the gate size that the power benefit is limited. However, face-to-face 3D ICs offer only slightly larger via sizes than monolithic 3D, and can also be considered fine-grained. Therefore, this chapter provides results on both face-to-face and monolithic 3D integration.

This chapter first provides a routing congestion aware physical design framework that modifies existing 2D placement engines for M3D placement, and also inserts MIVs into the layout. Next, it discusses how commercial 2D engines can be used for M3D placement, taking full advantage of state-of-the-art power and timing optimization techniques. Finally, it discusses how to partition the gates in the design such that voltage-drop is minimized, with a minimal impact on the temperature of the 3D chip.

4.1 *Congestion-Aware Placement for Gate-level Monolithic 3D ICs*

This section first formulates the problem, and then discusses how existing 2D placers can be minimally modified for M3D placement. It then presents a congestion model, and uses it to derive a congestion-driven placement algorithm. Finally, it presents results that demonstrate the effectiveness and benefits of the proposed techniques.

4.1.1 Overall Design Flow

4.1.1.1 Problem Formulation

The *"Projected 2D HPWL"* is defined as the half perimeter wirelength (HPWL) of a monolithic 3D IC if all the gates are projected onto a single placement layer. The total routing overflow is defined as the sum of routing demand minus routing supply on all global routing edges that are congested. The problem to be solved can then be stated as: *Given an initial monolithic 3D placement, repartition the gates with minimal change to the projected 2D HPWL, such that the total routing overflow is minimized*.

However, this formulation still requires an initial monolithic 3D placement. Therefore, the following problem is also solved: *Generate a 2D design, using minimally modified 2D tools, such that it represents a monolithic 3D IC with all the gates projected to a single tier.* If such a design is generated, then tier partitioning can directly be applied on top of it.

4.1.1.2 Design Flow

An overview of the proposed flow is shown in Figure 48. In this figure, the red boxes indicate steps that will be explained in detail in subsequent sections.

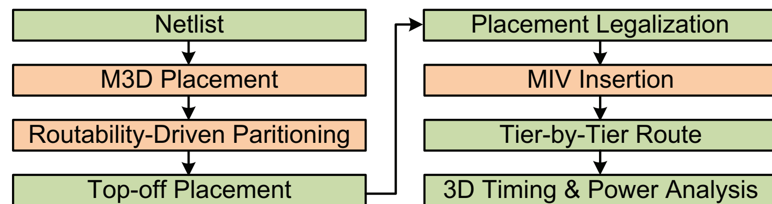


Figure 48: The design flow used for gate-level M3D placement.

From the synthesized netlist, an initial monolithic 3D IC placement result is obtained. Next, routability-driven partitioning is performed, which takes the initial placement solution and re-partitions the gates to improve the routed wirelength of the design. A top-off placement step is then performed to make sure that each tier in the monolithic 3D IC meets target density requirements. The last step in the placement process is legalization, which snaps the cells to the placement grid. Once the locations of cells are determined, MIVs need to be inserted into the whitespace between cells. MIVs can then simply be treated as I/Os in each tier, and a tier-by-tier route can be carried out using commercial tools (Cadence Encounter). Finally, parasitics are extracted tier-by-tier, and a separate parasitic file to represent MIV parasitics is created. All this information is fed into Synopsys PrimeTime to obtain 3D timing and power numbers.

4.1.2 Monolithic 3D IC Placement

This section first presents prior work in TSV-based 3D IC placement, and discusses why those approaches are not applicable to monolithic 3D ICs. Next, a methodology is proposed based on modifications to 2D IC tools. Finally, handling pre-placed memory macros in a 3D design while still using 2D IC tools is discussed.

The monolithic 3D gate-level placement problem is similar to the TSV-based problem, except that the via count need not be minimized. The first approach to TSV-based 3D placement is folding-based [13]. This takes an existing legal 2D placement, and transforms it to 3D by several folding operations. This approach generates inferior quality solutions [12], and is also not capable of handling pre-placed memory. The next method is partitioning-based [28], where the netlist is first partitioned and all tiers are placed simultaneously. Lastly, true 3D placement approaches exist [12, 21], where the half-perimeter wirelength (HPWL) is minimized in the x, y and z dimensions. However, in monolithic 3D ICs, the z dimension is so small ($1 - 2\mu m$) that attempting to minimize the z HPWL is not really necessary. In addition, all of these engines are geared towards TSV-based 3D, and try to

minimize the via count. This section demonstrates the fact that since monolithic vias are so small, only a minimally modified 2D placement engine suffices, and separate 3D placement engines are not required.

4.1.2.1 Placement-Aware Partitioning

An illustration of the proposed method for a two-tier monolithic 3D IC is shown in Figure 49. If the width and height of a 2D IC are W_{2D} and H_{2D} respectively, the M3D outline is defined such that the width and height of a 2D chip are divided by $\sqrt{2}$. This modification leads to exactly half the footprint of a 2D IC. All 2D placement engines have the concept of *chip capacity* (or target density), which is the maximum number of standard cells that can be placed in a given area. Since all the gates need to fit into half the area, simply doubling the capacity of the chip will work. *Any* existing 2D placer can be modified for this purpose, and this section implements a custom implementation of KraftWerk2 [59]. Clearly, the HPWL obtained after such a placement represents the HPWL of a monolithic 3D IC where all the tiers have been projected onto a single tier – the projected 2D HPWL.

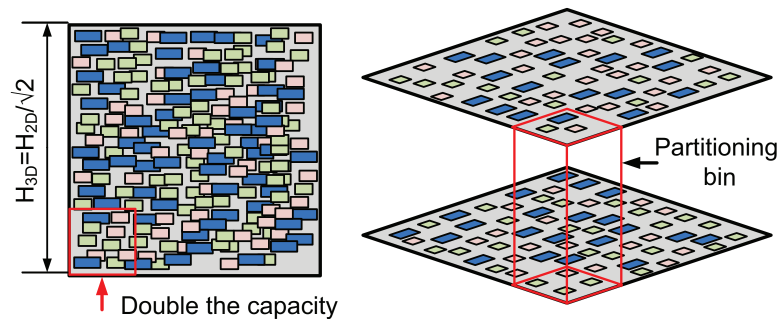


Figure 49: Placement-aware partitioning. A modified 2D engine is used to place all the gates into half the area, and then partitioned with area balance in each bin.

The next step is to partition the gates such that each tier has an equal number of gates, and the deviation from the initial (x, y) location is minimized. An obvious approach to partitioning the gates is a min-cut approach, and modifying the Fiduccia-Mattheyses [16] (FM) min-cut partitioner is straightforward, an overview of which is given below.

First, partition bins are defined in a regular fashion. Next, the design is partitioned

such that the cells in a given bin in the modified 2D result remain in the same bin after splitting. As will be discussed in Section 4.1.5.1, the choice of bin size affects solution quality greatly. This is because after partitioning, although each bin in each tier will contain the correct number of cells, these cells may not be distributed uniformly throughout the bin. If the partitioning bin size is much larger than the global placement bin size, there could potentially be large areas of extra-dense cell placement and large areas of whitespace. Therefore, top-off placement becomes necessary to obtain an acceptable placement solution that meets target density within each global bin.

Initially, a random, area-balanced (within each partition-bin) solution is created. The gain of a cell is defined as the reduction in the cutsize if the cell's tier is changed. A cell is "legal" if moving it does not violate the area-balance constraints within its partition bin. While moving a single cell from one tier to another will not affect the area balance too much, this condition ensures that too many cells are not moved from one tier to another.

Initially, all the cell gains are computed and stored in a bucket structure. All the cells also marked as "unlocked". Among all legal cells, the one with the highest gain is picked, moved to the other tier, and locked. Once a cell is moved, only the gains of its neighbors (connected by a net) needs to be updated. This process is continued until all the cells are locked. This is termed a *pass*. Several passes are performed until no more cutsize gains are achieved. Due to the nature of the incremental gain update, this algorithm runs in $O(C)$ time, where C is the number of cells. While the min-cut is straightforward, MIVs are extremely small and there is no real need to perform a min-cut on the netlist. Additional MIVs can be tolerated, if there is good reason to use them. A routability-driven partitioner is presented in Section 4.1.3, where additional MIVs are utilized to reduce routing congestion, and hence, routed WL.

Note that while this approach may appear somewhat similar to the local stacking transformation (LST) presented in [13], it is superior in one major aspect – the handling of pre-placed memory macros. The LST method obtains the initial (x, y) locations of all the cells

by scaling them from a *legal* 2D placement, and hence has no way to handle pre-placed memory macros in a 3D space. Handling them in the proposed method is straightforward, and will be discussed in the following subsection.

4.1.2.2 Handling Memory Macros

In a M3D design, hard macros such as memory are bound to be pre-placed. This section discusses how to handle these memory macros while still leveraging 2D IC tools. Let t_d be the target density required in the final, post-partitioned M3D design, and t'_d be the target density in the modified 2D placement. Consider the pre-placed memories in both tiers as shown in Figure 50(a).

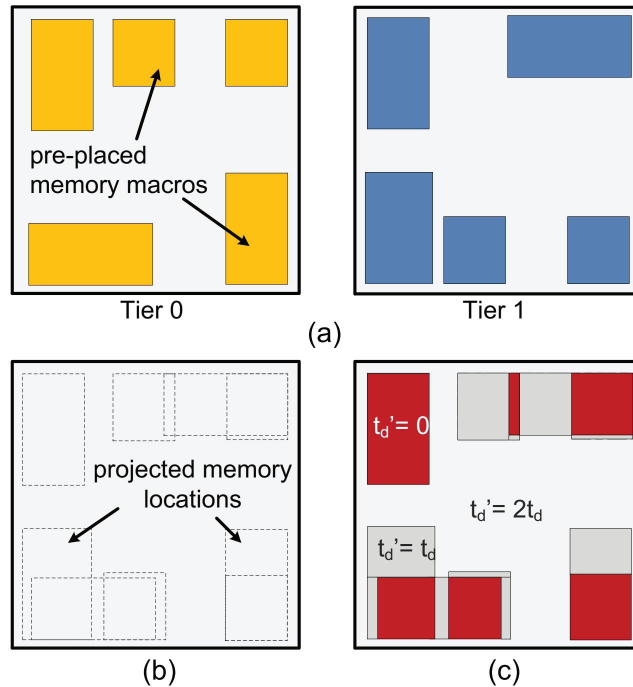


Figure 50: Handling pre-placed memory macros (a) Initial pre-placed locations, (b) Projection of both tiers onto the same plane, and (c) Modifying the target density to represent memory locations. t'_d is the target density in the modified 2D placement and t_d is the required target density in the final M3D design.

First, both these tiers are projected onto the same plane as shown in Figure 50(b). Those regions that have two memories overlapping cannot contain cells in any tier, and hence will have $t'_d = 0$. Those regions that have only one memory can contain cells in the tier where

the memory is not placed. To reflect this fact, the target density in those regions will not be doubled, or $t'_d = t_d$, as shown in Figure 50(c). Finally, the regions not containing memory will have cells of both tiers placed, and hence $t'_d = 2t_d$.

Handling these region-specific target density constraints is straightforward in the Kraftwerk placement system. In order to remove overlap between cells, it maintains a supply/demand system of placement space. The chip is divided into fine mesh tiles, and each mesh tile has a supply t_d. Each cell has demand 1 on each mesh tile that it occupies. Solving the poisson equation of supply minus demand gives the direction and amount to move each cell in order to equalize supply and demand. In this system, the supply of each fine mesh tile is set to t_d or $2t_d$ depending on requirements.

The partitioning process can also be modified easily. The regions with memory overlap in both tiers do not have cells, and need not be partitioned. Those cells placed in the regions with a single memory macro are moved to the tier not containing memory. Finally, the regions with cell overlap are partitioned as usual.

4.1.3 Routability-Driven Partitioning

The first step in building a routability-driven partitioner is to estimate the routing congestion in the monolithic 3D IC. The routing congestion is measured as the total routing overflow, which is the routing demand minus routing supply on all the global routing edges in the chip. The routing supply is determined from the number and pitch of metal layers, and this section discusses how to determine the 3D routing demand. This section then describes how to re-partition the monolithic 3D IC to reduce routing congestion.

4.1.3.1 Prior Work

While this is the first work to discuss a monolithic 3D routing demand model, this topic has been explored extensively for 2D ICs. The first approach is a grid-less approach [58] where the demand of a net is assumed to be distributed evenly along all possible Steiner tree combinations. This was extended to consider the differences between horizontal and vertical

segments in [24]. These approaches are more suitable for routability-driven placement, not partitioning, as both these papers try to minimize the overlap of the net bounding boxes. The other approach is to first decompose multi-pin nets into two pin nets, and add each two pin net into the demand estimate. The demand of each two pin net can be estimated either by maze routing [30], rough global (LZ) routing [37], or probabilistically [5]. This project chooses a probabilistic demand model because (1) It is extremely fast unlike maze routing, and (2) The predicted demand numbers are independent of net ordering unlike LZ routing. The first property is necessary as several solutions will be evaluated during partitioning, and the second property is essential for a partitioner as each re-compute of the demand of the same two-pin net must yield the same result.

4.1.3.2 Decomposing Multi-Pin Nets into Two-Pin Nets

This section presents a method of decomposing multi-pin nets into two-pin nets by constructing 3D rectilinear Steiner trees (RSTs). Currently, no tool exists to efficiently compute a 3D RST, so the net is projected to 2D, a 2D rectilinear Steiner minimum tree (RSMT) constructed, and then expanded back to 3D.

Sample points to be routed are shown in Figure 51(a). The points are first projected to a 2D plane, and a 2D RSMT is constructed using FLUTE [10] (Figure 51(b)). Now, while expanding this 2D RSMT to a 3D RST, the tiers of all the fixed points are already known. The tier of each Steiner point is determined by a majority vote of the tier of all of its neighbors. Any ties are broken in any arbitrary, deterministic manner. A neighbor is defined as any point (steiner or fixed) that the current Steiner point is connected to. If a neighbor does not have a tier determined yet, it is ignored during the current iteration of the majority vote operation. For example, when the 2D RSMT of Figure 51(b) is expanded, the tiers of the three steiner points that are connected to the fixed points are determined first. They each have two neighbors in one tier, and one undetermined neighbor. Therefore, they all lie in the same tier as the fixed points that they are connected to. Next, the tier of the

middle steiner point can be determined as the top tier as it has two neighbors in the top tier and one in the bottom tier. The resulting 3D RST is shown in Figure 51(c).

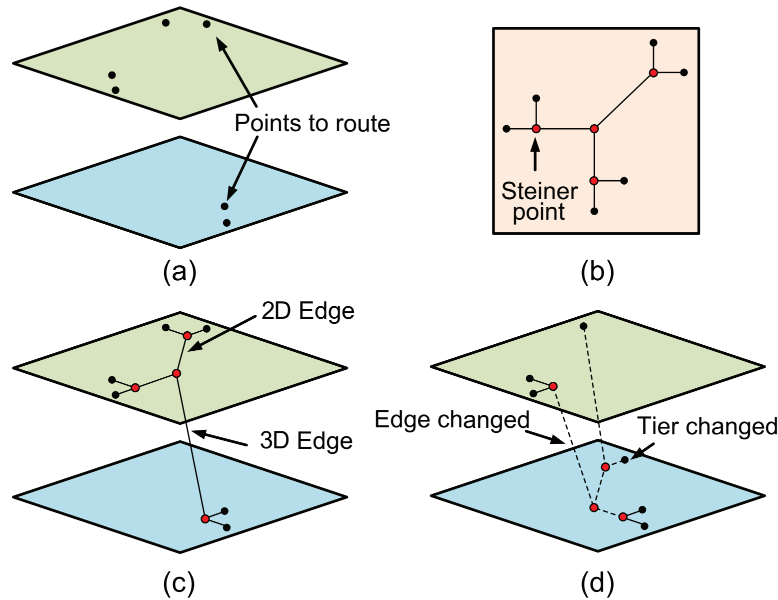


Figure 51: Construction of a 3D RST. (a) The points to be routed. (b) Project to 2D and construct a 2D RSMT. (c) Expand the 2D RSMT to a 3D RST. (d) If a cell changes tier, the 2D RSMT can be re-used.

Since the target is move-based partitioning, the change in topology needs to be quickly evaluated if the tier of a given cell is changed. Since such a change does not change the x & y co-ordinate of the cell, the same 2D RSMT can be reused. The tier of one cell is changed and the resulting 3D RST is shown in Figure 51(d). The expansion from Figure 51(b) is redone, and only the quick majority vote operation needs to be performed on the Steiner points. Note that the steiner point connected to the cell that has changed tier now has an equal number of neighbors in each tier. This tie can be broken in any deterministic manner, and this project always goes with the lower tier. Since the middle steiner point now has two neighbors in the bottom tier and one in the top tier, it is also assigned to the bottom tier.

As seen from this figure, a lot of the routing demand on the top tier is offloaded to the bottom tier, with an unchanged 3D bounding-box. Therefore, to evaluate the change in demand if the tier of a given cell is changed, the following steps need to be performed: (1) Redo the majority vote operation for all nets connected to that cell, (2) Delete the old

topology (rip-up) of the changed two-pin nets from the demand estimate, and (3) Add the new topology (re-route) of the changed two-pin nets into the demand estimate. Handling each two-pin net is described next.

4.1.3.3 3D Demand Model for Two-Pin Nets

A 3D routing graph is maintained for the entire chip. This section considers only that sub-graph that a given two-pin net spans. Although the focus is only on two tier monolithic 3D ICs, the model presented in this section is general, and is applicable to any number of tiers. An example $4 \times 3 \times 2$ three-tier subgraph is shown in Figure 52.

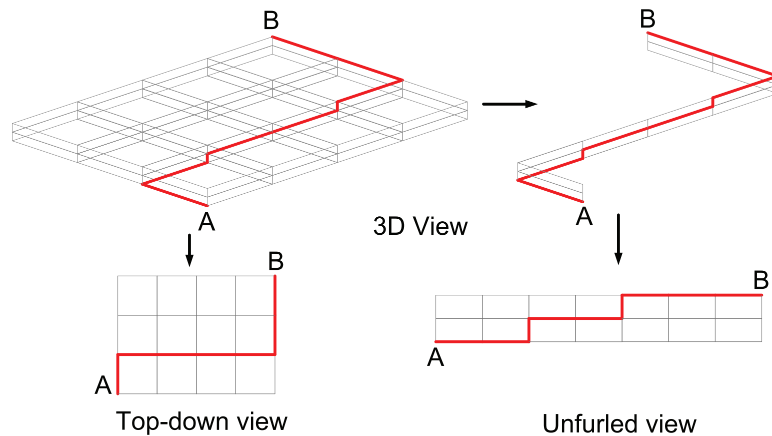


Figure 52: A legal route from A to B in a $4 \times 3 \times 2$ grid. The top-view is limited to two bends, while the unfurled view can have unlimited bends.

Assume that the net (A-B) spans a $l \times m \times n$ routing sub-graph. The probabilistic routing demand contributed by this two-pin net on each edge within this sub-graph needs to be computed. One possible route from A to B is highlighted in red. Many such legal routes exist, and a probabilistic demand model assumes that each $legal$ route is equally probable. Therefore, the key to such a demand model is to correctly identify which routes are legal.

Two key observations that help derive the demand model are: (1) Looking at the 3D demand graph from the top-view, each bend represents the usage of a local via. Since current global routers try to minimize the usage of local vias, this is limited to at most two

bends (or local vias) in the top view [5, 37]. (2) A new view called the unfurled view is defined, which unfurls the routing graph along a legal route (refer Figure 52). In such a view, movement along either x or y directions look the same. In this view, irrespective of the number of bends, the number of MIVs is always the same and equal to exactly $n-1$. For example, in Figure 52, two MIVs always connect A and B, irrespective of the number of bends in the route. Therefore, there are no limits to the number of bends in the unfurled view.

Assuming the above constraints, the total number of routes from A to B is $(l+m)\times^{(l+m+n)} C_n$. First, given the top-view constraint, the sum of all the probabilities along all the edges that look identical in the top-view is given by:

$$\sum_{i=1}^{n} P_{(x,x+1),y,i} = \frac{1}{l+m} \times \begin{cases} (l-x), & \text{if } y=0 \\ (x+1), & \text{if } y=m \\ 1, & \text{otherwise} \end{cases} \tag{28}$$

A similar expression can also be written for all the y edges. Next, in the unfurled view, all edges with the same $(x+y)$ look the same. Therefore, let i represent $(x+y)$. Since there is no limit to the number of bends, the routing probability on any horizontal edge is given by a uniform probability distribution:

$$P_{(i,i+1),z} = \frac{^{(i+z)}C_i \times^{(l+m+n-i-z-1)} C_{(l+m-i-1)}}{^{(l+m+n)}C_n} \tag{29}$$

Equations (28) & (29) can be combined to give the routing probability on any x edge in the 3D graph:

$$K_{3D} = \frac{^{(x+y+z)}C_z \times^{(l+m+n-x-y-z-1)} C_{(l+m-x-y-1)}}{(l+m)\times^{(l+m+n)} C_n}$$

$$P_{(x,x+1),y,z} = K_{3D} \times \begin{cases} (l-x), & \text{if } y=0 \\ (x+1), & \text{if } y=m \\ 1, & \text{otherwise} \end{cases} \tag{30}$$

A similar expression can also be computed for all the y edges. Once the probabilities of the x & y edges have been computed, the probability on each z edge can be computed by visiting them in turn, and setting the probability to be the sum of the probability on all incoming edges (towards A) minus the sum of the probability on all the outgoing edges (towards B).

4.1.3.4 Interdependent Supply/Demand Model

In 2D ICs, there are two types of tracks – x and y. Using an x track does not affect the supply of y tracks, and vice-versa. In monolithic 3D ICs, the number of z tracks available also needs to be taken into account. These z tracks, however, are not independent of the x and y track usage. Assuming that the top metal layer is vertical, this fact is illustrated in Figure 53. This figure shows the top view of the top metal layer of one global routing bin. The green squares represent potential MIV landing pad sites whose pitch is determined by the pitch of the top metal layer.

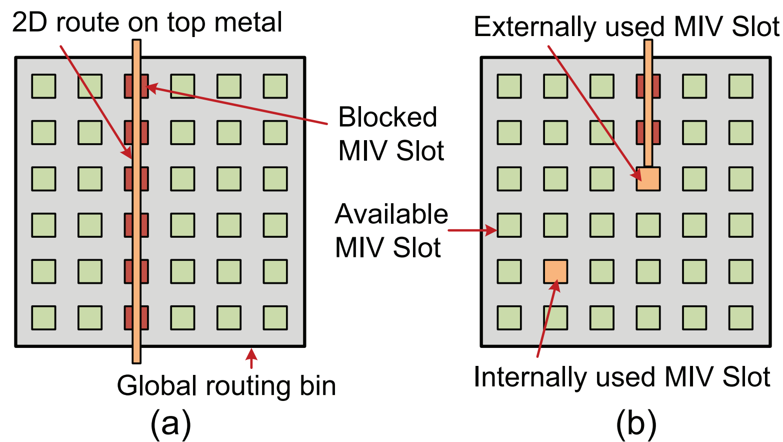


Figure 53: A view of the top metal layer that contains MIV landing pads. (a) A 2D wire on the top metal layer blocks potential MIV landing pad slots. (b) If MIVs connect to cells outside the current bin (external), they block other MIVs. If MIVs connect to cells within the current bin (internal), they do not block other potential MIV slots.

There are three effects that need to be modelled. First, assume that a 2D wire on the top metal layer crosses this bin. As shown in Figure 53(a), this 2D route blocks potential MIV landing pad sites, and hence reduces the 3D supply. Next, as shown in Figure 53(b), if a

MIV lands on the top metal layer (from the other die), and continues onto a different global routing bin, this is termed an externally used MIV slot. Such connections use one MIV slot, but also block others. Finally, if an MIV lands on the top metal layer but connects to a gate within the same bin itself, it is termed an internally used MIV. As seen from this figure, it uses one MIV slot but does not block other MIV slots. However, this requires an entire via stack from the top metal to the lowest metal to connect to the cell. This via stack causes via blockages [8], which reduces the 2D supply in the lower metal layers.

Let W_B and H_B be the width and height of the global routing bin. N_{MH} and N_{MV} are the number of horizontal and vertical metal layers, respectively. Let P_{Hi} and P_{Vi} be the pitch of the ith horizontal and vertical metal layer respectively. Note that M1 is ignored as it is usually used for within-cell routing. Therefore, the "first" metal layer is actually M2. Also, this section assumes that the top metal layer has a preferred vertical direction. The derivation can also easily be carried out if it is horizontal.

If the top metal pitch is assumed to be the only factor determining the number of MIV slots, then the number of vertical and horizontal MIV slots are: $N_H = W_B/P_{N_{MV}}$ and $N_V = H_B/P_{N_{MH}}$. However, not all these slots are accessible. This is because each metal layer only contributes a finite number of tracks that can connect to MIVs in this bin. The number of MIV slots can then be given as

$$N_{MIV} = 2N_H N_{MV} + 2N_V N_{MH} - 4N_{MV} N_{MH} \tag{31}$$

This can then be divided into a matrix with N'_H and N'_V effective horizontal and vertical slots. It should be noted that this routing-based constraint on the number of MIVs is far more restrictive than computing the number of MIVs slots available by simply looking at the whitespace available for MIV insertion. It can be shown that even if all the above MIV slots are utilized, it will occupy only $2-3\%$ of the area of a given placement bin.

Next, to determine the number of blocked MIV slots, the number of 2D and 3D routes that use the top metal layer needs to be determined. This requires metal layer assignment, which is a complicated problem. Instead, the routes are assumed to be assigned to metal

layers based on the inverse ratio of pitch, i.e., a larger pitch metal will have fewer wires. Let $N_{N,2D,i}$ be the number of 2D routes that cross the north edge on metal layer i. Similar definitions can be made for 3D routes and the east, west, and south edges. Let $N_{N,2D}$ be the total number of 2D routes crossing the north edge, and P_i be the pitch of the ith metal layer. For each vertical metal layer i, $N_{N,2D,i} = N_{N,2D}/(P_i.\sum_j(1/P_j))$. It is pessimistically assumed that any 2D or 3D wire crossing an edge goes all the way to the center of the bin. The number of blocked MIV slots (assuming the top metal is vertical) can then be given as

$$\begin{aligned} N_{MIV,Blk} =& 0.5N'_V(N_{N,2D,N_{MV}} + N_{S,2D,N_{MV}}) \\ &+(0.5N'_V - 1)(N_{N,3D,N_{MV}} + N_{S,3D,N_{MV}}) \end{aligned} \tag{32}$$

The first term in the above equation represents the number of MIV slots blocked by 2D wires and the second term represents the number of MIV slots blocked by external MIV connections. The actual number of MIV slots can be obtained by subtracting Equation (32) from Equation (31).

The next step is to calculate the 2D supply reduction due to the via blockages introduced by MIV connections. Let $N_{int,3D}$ be the number of internal MIV connections in this bin. Each bin is divided into four quadrants, numbered one through four, in the usual naming convention. The number of vias in the first quadrant, on metal layer i, can then be given as

$$N_{via,1,i} = 0.25N_{int,3D} + \sum_{j<i} 0.5(N_{N,3D,j} + N_{E,3D,j}) \tag{33}$$

If $W_{via,i}$ is the width of the via on metal layer i, then the fraction of metal layer i in the first quadrant that is blocked by vias is given as [8]:

$$B_{via,1,i} = \sqrt{\frac{N_{via,1,i}(W_{via,i} + 0.5P_i)}{0.25W_B H_B}} \tag{34}$$

Based on this, the actual 2D supply on the north edge of the routing bin is given as

$$SUP_N = W_B \sum_{i=1}^{N_{MV}} (1 - 0.5(B_{via,1,i} + B_{via,2,i}))/P_i \tag{35}$$

Similar expressions can then be derived for all the other edges as well.

4.1.3.5 *Min-Overflow Partitioning*

Routability-driven (min-overflow) partitioning can now make use of the 3D demand model. First, min-cut partitioning as described in Section 4.1.2 is performed. A min-overflow partitioning is then performed on top of this solution. Total overflow is used as the metric to be minimized, which is defined as the summation of the overflow on all the 2D and 3D edges in the chip that are congested. The overflow-gain of a cell is then the reduction in the total overflow when its tier is changed, and it is computed by the procedure outlined in Subsection 4.1.3.2.

Let C be the set of all cells and N be the set of all the nets in the design. In the min-cut partitioner, once a cell is moved, only the gains of its neighbors needs to be updated. However, the overflow depends on all nets that use a particular routing edge, not just those connected to this cell. If a cell is moved, it affects several routing edges. Any other net that uses the affected routing edges will now need to have its overflow updated. Since the gain is defined for moving a cell, all cells connected to such nets will also need to have their gain updated. For cells connected to nets with large bounding boxes, up to C cells will need to be updated every time it is moved. This means that maintaining a priority queue with all cells, such as in the default FM algorithm, would lead to a time complexity of O(C^2). This neglects the time necessary to rebuild the queue, which adds a further $O(log(C))$ complexity. Overall, this would lead to excessively large runtime, making it infeasible. A heuristic that reduces the time complexity significantly is now presented, and shown in Algorithm 3.

The top-level function in this algorithm is $MinOverflow()$. Initially, the demand estimate is cleared i.e, all nets are removed, and the utilization on each routing edge is set to 0. Next, there are two stages, build and refine, both of which are similar, and handled by the $Stage()$ function. In the build phase, all the nets are initially set to invalid. In both stages, the nets are then sorted by bounding-box. This is because nets with a larger bounding box have a greater impact on the routing graph, and will be processed first. During the build

Algorithm 3: A Min-Overflow Partitioning Heuristic

```
1  Function MinOverflow()
2      demandEstimate →Clear() ;
3      Stage(build) ;
4      Stage(refine) ;
5  end
6  Function Stage(type)
7      if (type == build) then
8          ∀n ∈ N : n →valid = false ;
9      end
10     Sort N in descending order of bounding-box ;
11     foreach n ∈ N do
12         if (type == build) then
13             demandEstimate →AddRST(n →rst) ;
14             n →valid = true ;
15         end
16         FM( n → c_n ) ;
17     end
18 end
```

phase, the 3D-RST of the net currently being processed is added into the demand estimate, and the net is set to valid. Next, irrespective of stage, the $FM()$ function (to be described later) is performed on the cells of the current net. Note that in the build phase, the demand estimate does not have all the nets included, only the ones that have been processed so far. This is to avoid any noise introduced by a bad initial random partitioning of the unprocessed nets.

The $FM()$ function is similar to the basic algorithm described in Section 4.1.2, with a few differences: (1) A heap is used instead of a bucket, as the gains are not integer values. (2) Only a subset of cells that belong to a given net are considered, (3) When a cell is moved to another tier, the gains of all cells within the current subset are updated, and (4) The gain function is the global max-overflow gain, considering all "valid" nets in the design, not just the current net being processed.

The above heuristic adds one net at a time into the demand estimate, maintaining a local optima of the global total overflow after each net is added. Once all the nets are

added, each net is processed again to further reduce the overflow. This approach leads to a time complexity of $O(N.(rms_{N_d})^2)$, where rms_{N_d} is the root-mean-square of the net degrees. This value does not scale much with circuit size, and therefore, the heuristic is more or less linear in runtime.

4.1.4 Router-based 3D-Via Insertion

To continue with the P&R flow, routing and then parasitic extraction needs to be performed. However, current routers can only handle 2D ICs, and the usual approach is to split the 3D design into separate designs for each tier, each of which can be routed independently. This requires the locations of the MIVs to be known, so that they can be represented as I/O pins within each tier.

Once the partition of all cells are finalized, current TSV-based placers perform a TSV and cell co-placement step [28, 12] to determine the via locations. However, MIVs are so small that they can actually be handled by the router, and the only hurdle is the lack of an existing 3D commercial router. However, 2D commercial routers are capable of routing to pins on different metal layers, and a method to trick existing 2D commercial routers into performing MIV insertion is illustrated in Figure 54.

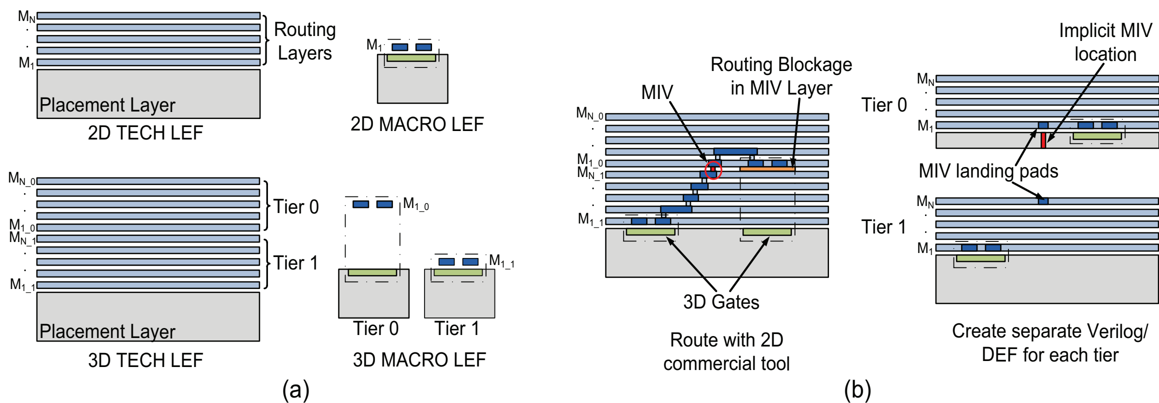
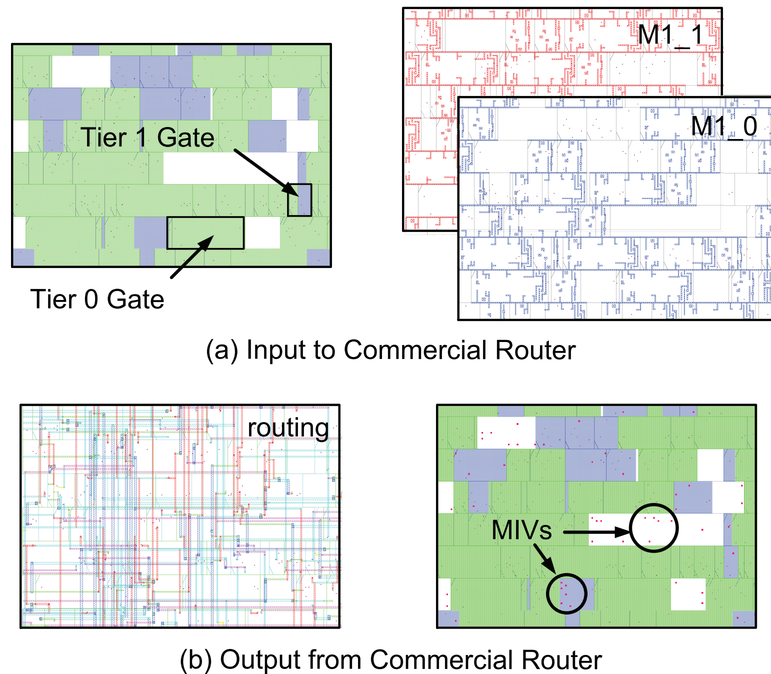


Figure 54: An overview of the router-based MIV insertion methodology. (a) The technology and macro LEF are modified to represent a two-tier monolithic 3D IC. (b) The structure that is fed into the commercial router, which is then routed. The MIV locations are extracted and separate verilog/DEF files are created for each tier.

First, all the metal layers in the technology LEF are duplicated to yield a new 3D LEF with twice the number of metal layers. Next, for each standard cell in the LEF file, two flavors are defined – one for each tier. The only difference between the two flavors is that their pins are mapped onto different metal layers depending on its tier. Next, each cell in the 3D space is mapped to its appropriate flavour, and forced onto the same placement layer. Note that this will lead to cell overlap in the placement layer, but there will be no overlap in the routing layers (Figure 55). Routing blockages are placed in the via layer between the two tiers, to prevent MIVs over cells. This structure is then fed into an existing commercial router (Cadence Encounter). Once routed, the routing topology is traced to extract the MIV locations, and separate verilog/DEF files are generated for each tier.



(a) Input to Commercial Router

(b) Output from Commercial Router

Figure 55: Screenshots of router-based MIV insertion (a) All the gates are placed in the same placement layer, but no overlap exists in the routing layers. (b) The result after routing. The MIV locations are highlighted in red.

4.1.5 Experimental Results

Eight benchmarks are chosen, six of which are from the OpenCores benchmark suite. In addition, two processor designs, the OpenSPARC (OS_T2) and LEON3 cores are chosen.

These designs vary in size from a few tens of thousands of gates to half a million gates. They are synthesized with a 28nm cell library, and their statistics are tabulated in Table 20. Of these eight designs, three have memory macros, as listed under the memory area column in Table 20.

Table 20: The various benchmarks considered in this section.

Circuit	Clock Period (ns)	#Cells	#Nets	Area (mm^2)		#ML
				Std. Cell	Memory	
mul_64	1.2	21,671	22,399	0.078	0	4
LEON3	0.9	17,419	19,069	0.051	0.034	4
nova	2.3	57,339	60,867	0.179	0.028	6
rca_16	0.4	67,086	75,786	0.263	0	4
aes_128	0.5	133,944	138,861	0.349	0	5
jpeg	1.5	193,988	238,496	0.739	0	4
OS_T2	1.5	316,573	334,374	1.110	0.468	6
fft_256	1.0	488,508	492,499	1.833	0	5

In addition to the clock period, number of cells, and number of nets, this table also shows the minimum number of metal layers with which the 2D placement is routable. This is used as the number of metal layers for both 2D and monolithic 3D versions of each design. The footprint area of each design is chosen such that the standard cells have a target density of 70%. All monolithic 3D designs are implemented such that they have exactly 0% area overhead compared to their corresponding 2D version, i.e., exactly 50% footprint area, *irrespective of MIV count*. This condition also ensures that the standard cells in the M3D design have a target density of 70%. The diameter of each MIV is assumed to be $100nm$, with a resistance of 2Ω and a capacitance of $0.1fF$ [33].

In order to obtain pre-placed memory macro locations for 3D, the memory macros are partitioned architecturally. An example of this for the OS_T2 benchmark is shown in Figure 56. The 2D design contains several modules such as load-store unit (lsu), instruction-fetch unit (ifu) e.t.c. Roughly half the memories in each module are allocated to each tier, and the memories are manually placed to mimic the 2D placement as close as possible.

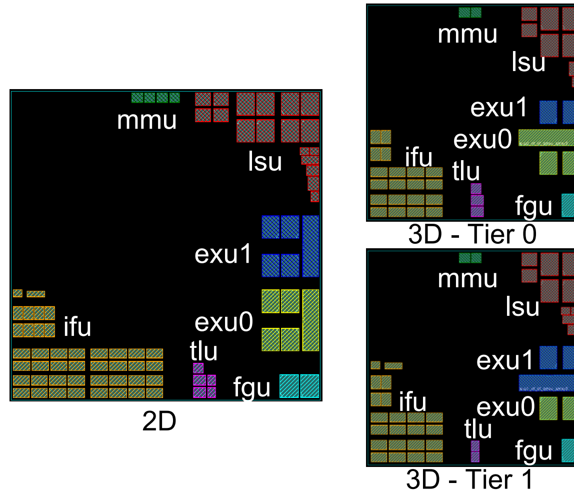


Figure 56: Manual partitioning of the memories in the OS_T2 benchmark. The memories belonging to each sub-module are partitioned, and placed in a configuration similar to that in 2D.

4.1.5.1 The Impact of Partitioning Bin Size

As discussed in Subsection 4.1.2.1, the choice of partition-bin size affects the solution quality greatly. From the perspective of cell displacement, smaller bin sizes are better. However smaller bin sizes mean more partitioning-bins, which leads to more area-balance constraints that the partitioner needs to satisfy. More constraints imply a worse objective function, which means a larger cutsize in the min-cut partitioner. Since routed WL is more important than 3D HPWL, more 3D vias mean that an appropriate whitespace location needs to be found for more MIVs, which may not always be feasible. Therefore, a smaller bin size may not always lead to lower wirelength. To quantify this effect, the min-cut partitioner is run on all benchmarks with varying bin sizes, and results are tabulated in Table 21.

For each benchmark, five different bin sizes are evaluated. The MIV count after router-based MIV insertion and the projected 2D HPWL which is the objective function of the top-off placement are tabulated. As expected, increasing the bin size always reduces the MIV count due to the partitioner having more freedom, and also always increases the projected 2D HPWL as the final (x, y) location of cells deviates more. However, the impact on routed

Table 21: The impact of partition bin size on solution quality.

mul_64

Bin W (μm)	#MIV ($\times 10^3$)	Proj. 2D HPWL (m)	Routed WL (m)	PDP (mW-ns)
5	15.41 (1.00)	0.31 (1.00)	0.46 (1.00)	35.61 (1.00)
10	8.35 (0.54)	0.31 (1.00)	0.45 (0.97)	34.99 (0.98)
20	5.67 (0.36)	0.32 (1.01)	**0.44 (0.96)**	**34.63 (0.97)**
40	4.73 (0.30)	0.32 (1.02)	0.45 (0.98)	35.22 (0.98)
80	3.50 (0.22)	0.34 (1.08)	0.47 (1.02)	35.34 (0.99)

aes_128

Bin W (μm)	#MIV ($\times 10^3$)	Proj. 2D HPWL (m)	Routed WL (m)	PDP (mW-ns)
5	95.43 (1.00)	1.94 (1.00)	3.00 (1.00)	105.16 (1.00)
10	63.75 (0.66)	1.97 (1.01)	**2.95 (0.98)**	**105.05 (0.99)**
20	56.63 (0.59)	2.02 (1.04)	2.99 (0.99)	105.37 (1.00)
40	35.96 (0.37)	2.27 (1.17)	3.19 (1.06)	107.04 (1.01)
80	16.76 (0.17)	2.43 (1.25)	3.34 (1.11)	108.48 (1.03)

LEON3

Bin W (μm)	#MIV ($\times 10^3$)	Proj. 2D HPWL (m)	Routed WL (m)	PDP (mW-ns)
5	12.50 (1.00)	0.36 (1.00)	0.54 (1.00)	25.92 (1.00)
10	6.79 (0.54)	0.37 (1.00)	0.53 (0.97)	25.60 (0.98)
20	5.77 (0.46)	0.37 (1.01)	**0.52 (0.96)**	**25.51 (0.98)**
40	5.44 (0.43)	0.37 (1.02)	0.53 (0.97)	25.62 (0.98)
80	4.19 (0.33)	0.38 (1.03)	0.53 (0.97)	26.04 (1.00)

jpeg

Bin W (μm)	#MIV ($\times 10^3$)	Proj. 2D HPWL (m)	Routed WL (m)	PDP (mW-ns)
5	161.06 (1.00)	3.79 (1.00)	5.40 (1.00)	359.20 (1.00)
10	88.84 (0.55)	3.78 (0.99)	5.32 (0.98)	352.72 (0.98)
20	56.79 (0.35)	3.83 (1.01)	**5.27 (0.97)**	**350.51 (0.97)**
40	47.29 (0.29)	3.90 (1.02)	5.30 (0.98)	351.06 (0.97)
80	35.47 (0.22)	4.14 (1.09)	5.48 (1.01)	355.50 (0.99)

nova

Bin W (μm)	#MIV ($\times 10^3$)	Proj. 2D HPWL (m)	Routed WL (m)	PDP (mW-ns)
5	44.81 (1.00)	1.27 (1.00)	2.09 (1.00)	68.84 (1.00)
10	25.66 (0.57)	1.27 (1.00)	2.01 (0.96)	68.08 (0.98)
20	22.25 (0.49)	1.29 (1.01)	**1.98 (0.94)**	**68.07 (0.98)**
40	17.07 (0.38)	1.30 (1.02)	1.99 (0.95)	67.38 (0.97)
80	14.34 (0.32)	1.35 (1.06)	1.99 (0.95)	68.44 (0.99)

OS_T2

Bin W (μm)	#MIV ($\times 10^3$)	Proj. 2D HPWL (m)	Routed WL (m)	PDP (mW-ns)
5	270.77 (1.00)	11.44 (1.00)	-	-
10	149.36 (0.55)	11.62 (1.01)	17.41 (1.00)	520.20 (1.00)
20	129.30 (0.47)	11.64 (1.01)	**17.36 (0.99)**	**517.50 (0.99)**
40	108.17 (0.39)	11.72 (1.02)	17.40 (0.99)	518.10 (0.99)
80	102.42 (0.37)	11.79 (1.03)	17.44 (1.00)	519.90 (0.99)

rca_16

Bin W (μm)	#MIV ($\times 10^3$)	Proj. 2D HPWL (m)	Routed WL (m)	PDP (mW-ns)
5	53.38 (1.00)	0.79 (1.00)	1.52 (1.00)	23.91 (1.00)
10	31.83 (0.59)	0.82 (1.03)	**1.50 (0.98)**	**23.76 (0.99)**
20	19.34 (0.36)	0.86 (1.08)	1.53 (1.00)	24.07 (1.00)
40	14.16 (0.26)	0.90 (1.13)	1.54 (1.01)	24.56 (1.02)
80	11.25 (0.21)	0.93 (1.16)	1.56 (1.02)	24.75 (1.03)

fft_256

Bin W (μm)	#MIV ($\times 10^3$)	Proj. 2D HPWL (m)	Routed WL (m)	PDP (mW-ns)
5	368.22 (1.00)	14.10 (1.00)	-	-
10	227.62 (0.61)	14.11 (1.00)	24.76 (1.00)	775.32 (1.00)
20	164.78 (0.44)	14.34 (1.01)	24.71 (0.99)	767.55 (0.99)
40	145.87 (0.39)	14.48 (1.02)	24.58 (0.99)	755.23 (0.97)
80	130.14 (0.35)	14.49 (1.02)	**24.17 (0.97)**	**752.00 (0.97)**

wirelength is mixed, which is due to the trade-off mentioned earlier. There is a clear sweet spot in terms of bin size. Increasing the bin size reduces the MIV count, which means that MIV insertion is easier, which reduces the routed wirelength. However, increasing the bin size too much means that the increase in projected 2D HPWL outweighs any benefits obtained from fewer MIVs. This sweet spot is different for different benchmarks, but Table 21 suggests that a bin size of $10 - 20\mu m$ works well across a wide range of designs, for this technology. Note that with a different technology, this bin size will need to change to keep the number of cells per bin a constant. Since sweeping the bin size is not feasible for each new benchmark, a partitioning bin size of $20\mu m$ is chosen for all benchmarks, and all subsequent results presented in this section assume this bin size.

4.1.5.2 Impact of Router-based MIV Insertion

The conventional method for 3D via insertion is to perform a post-place cell & 3D via co-placement [28, 12]. This section compares router-based MIV insertion scheme against this conventional technique. For reasons that will be given in Subsection 4.1.5.4, it is assumed that monolithic 3D has one metal layer removed from the top tier. Both placement-driven MIV insertion, as well as the proposed router-driven MIV insertion are performed, and results are tabulated in Table 22.

In this table, entries marked with a * indicate that that particular flavor is unroutable, and the wirelength reported is on designs with many thousands of DRC violations. Since reliable parasitic extraction cannot be performed on such designs, only wirelength and MIV count are compared. As observed from this table, the placement-driven MIV insertion often produces results that are unroutable. In those cases that are routable, router-based MIV insertion improves the routed WL by up to 15%. This is because the placement-based method tends to cluster vias together, leading to large clumps of vias, and large areas without any vias. When routing the placement-based method with the commercial router, no significant congestion is observed during the trial route or global route phase. However,

Table 22: The impact of router-based MIV insertion. Entries marked with a * are unroutable.

Circuit	Placement-driven		Router-driven	
	WL (m)	#MIV ($\times 10^3$)	WL (m)	#MIV ($\times 10^3$)
mul_64	0.530	3.723	0.473	5.677
LEON3	0.628	3.907	0.549	5.772
nova	2.170	13.687	2.031	22.256
rca_16	1.575	11.749	1.535	19.344
aes_128	3.213	35.026	2.988	56.632
jpeg	6.233	24.010	5.304	56.791
OS_T2	21.740*	73.805*	17.469	129.308
fft_256	31.829*	71.272*	25.133	164.784
Geo-Mean	3.348	17.859	2.943	31.489
Norm.	1.000	1.000	0.879	1.763

the vias are so small that it becomes difficult to route to them causing huge issues during detailed routing. The router-based method, although it has more MIVs (due to multiple vias inserted per net), spreads them out over the area of the chip, increasing the routability.

4.1.5.3 Impact of Routability-Driven Partitioning

Starting with the min-cut solution, routability-driven partitioning is performed with and without the interdependent supply/demand (IdS) proposed in Subsection 4.1.3.4. It is also assumed that one metal layer is reduced from the top tier in M3D. The supply, demand, and overflow of the min-cut partition of mul_64 with and without IdS is plotted in Figure 57. From this figure, it is seen that in the case of IdS, the supply of the MIV layer is reduced due to the demand in the tier 1 top metal, and vice-versa. Clearly, not considering IdS during min-overflow partitioning significantly overestimates the MIV supply. The results are tabulated in Table 23.

When compared with the min-cut solution, the min-overflow partitioner without IdS can reduce the routed WL by up to 4.30% (mul_64) and the PDP by up to 3.14% (fft_256). On average, the min-overflow partitioner without IdS gives 1.8% and 0.9% better wirelength and PDP respectively. If, however, IdS is considered during partitioning, up to a further

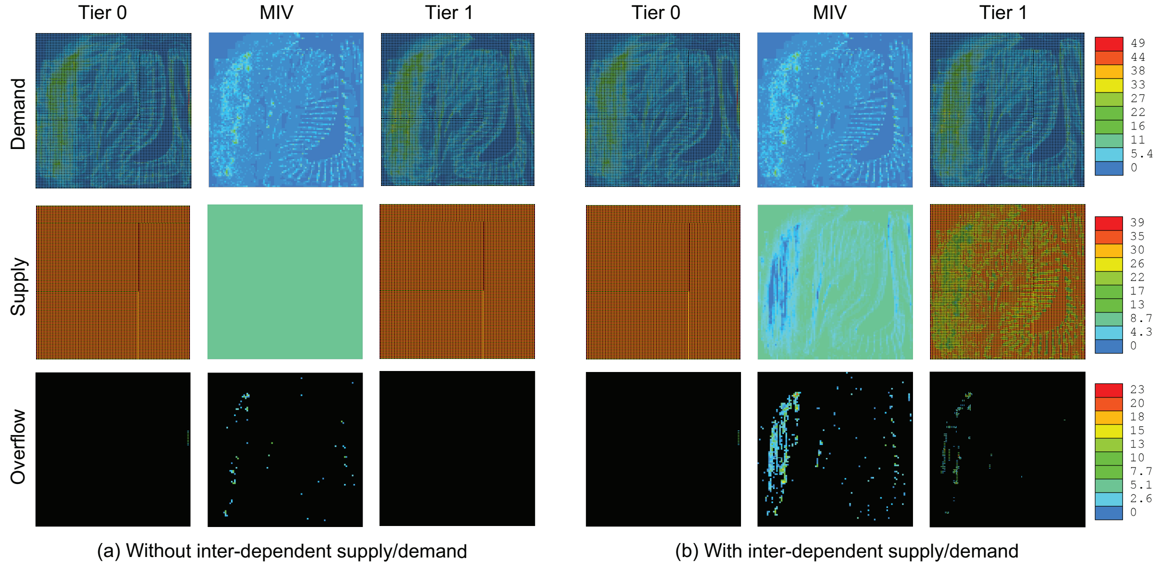


Figure 57: Supply, demand, and overflow maps of the mul_64 benchmark for min-cut based partitioning solution. If interdependent supply/demand is considered, a significant reduction in supply in densely wired areas is observed, leading to more overflow.

Table 23: The impact of routability-driven partitioning on monolithic 3D IC designs.

Circuit	Min-cut			Min-overflow (w/o IdS)			Min-overflow (with IdS)		
	WL (m)	PDP (mW-ns)	#MIV ($\times 10^3$)	WL (m)	PDP (mW-ns)	#MIV ($\times 10^3$)	WL (m)	PDP (mW-ns)	#MIV ($\times 10^3$)
mul_64	0.47	35.52	5.67	0.45	34.91	7.24	0.45	33.11	6.32
LEON3	0.54	25.95	5.77	0.55	26.26	6.69	0.53	25.86	5.84
nova	2.03	69.82	22.25	1.98	68.15	27.51	1.98	67.94	25.05
rca_16	1.53	23.75	19.34	1.50	23.58	25.82	1.49	23.44	23.14
aes_128	2.98	105.47	56.63	2.98	104.05	63.73	2.96	103.97	61.95
jpeg	5.30	351.93	56.79	5.27	357.75	72.43	5.18	349.53	63.10
OS_T	17.46	522.30	129.30	17.16	517.95	164.86	16.61	509.55	134.98
fft_256	25.13	791.64	164.78	24.18	766.72	222.05	23.26	758.79	180.66
Geo-Mean	2.94	111.25	31.48	2.89	110.22	39.41	2.84	108.47	34.58
Norm.	1.00	1.00	1.00	0.98	0.99	1.25	0.96	0.97	1.09

3.8% and 2.65% boost in the WL and PDP is obtained, respectively. In this case, the min-cut solution can be improved by up to 7.44% w.r.t. WL and 4.31% w.r.t. PDP. This takes the average WL and PDP gain over min-cut to 3.4% and 2.2%, respectively. In addition, the min-overflow solution without IdS underestimates the congestion in the MIV layer, and, on average uses 25.2% more MIVs than the min-cut solution. If IdS is considered during partitioning, the MIV count increase over min-cut goes down to 9.8%.

4.1.5.4 Reducing Metal Layers in Monolithic-3D

Cost is one of the primary concerns that needs to be addressed before 3D ICs can be widely adopted. If each tier in a monolithic 3D IC uses the same number of metal layers as 2D, the additional cost over 2D is the bonding of the empty silicon wafer. One method to offset the increased cost is to reduce the number of metal layers in 3D, reducing the total cost of the chip.

Reducing the number metal layers in monolithic 3D is now explored. The default case is when both tiers have the same number of metal layers as 2D (Table 20). Reducing one metal layer from the top-tier alone is termed “Tm1”, and reducing one metal layer from each of the top and bottom tiers is termed “Tm1_Bm1”. For each of these cases, min-cut partitioning, as well as min-overflow partitioning, with and without IdS is performed. The wirelength and PDP for all these cases is plotted in Figure 58. The curves for 2D are also plotted as a comparison.

The first thing observed is that even with a reduced metal count, all designs in monolithic 3D are able to be routed with zero DRC violations. These designs were not routable with fewer metal layers in 2D, so the fact that they are now routable indicates that monolithic 3D reduces the routing demand significantly. The next thing to note is that, as expected, reducing the metal layer count increases the wirelength and PDP. The magnitude of this increase depends on how congested the initial design is to begin with. In addition, the min-overflow partitioner helps both wirelength and PDP significantly. In many cases, the “Tm1” min-overflow (without IdS) result is better than the min-cut with all metal layers. Similarly, the addition of IdS into the partitioner gives a huge WL and PDP benefit. In several cases, designs can have two metal layers removed and still have lower WL than the min-cut case with all metal layers.

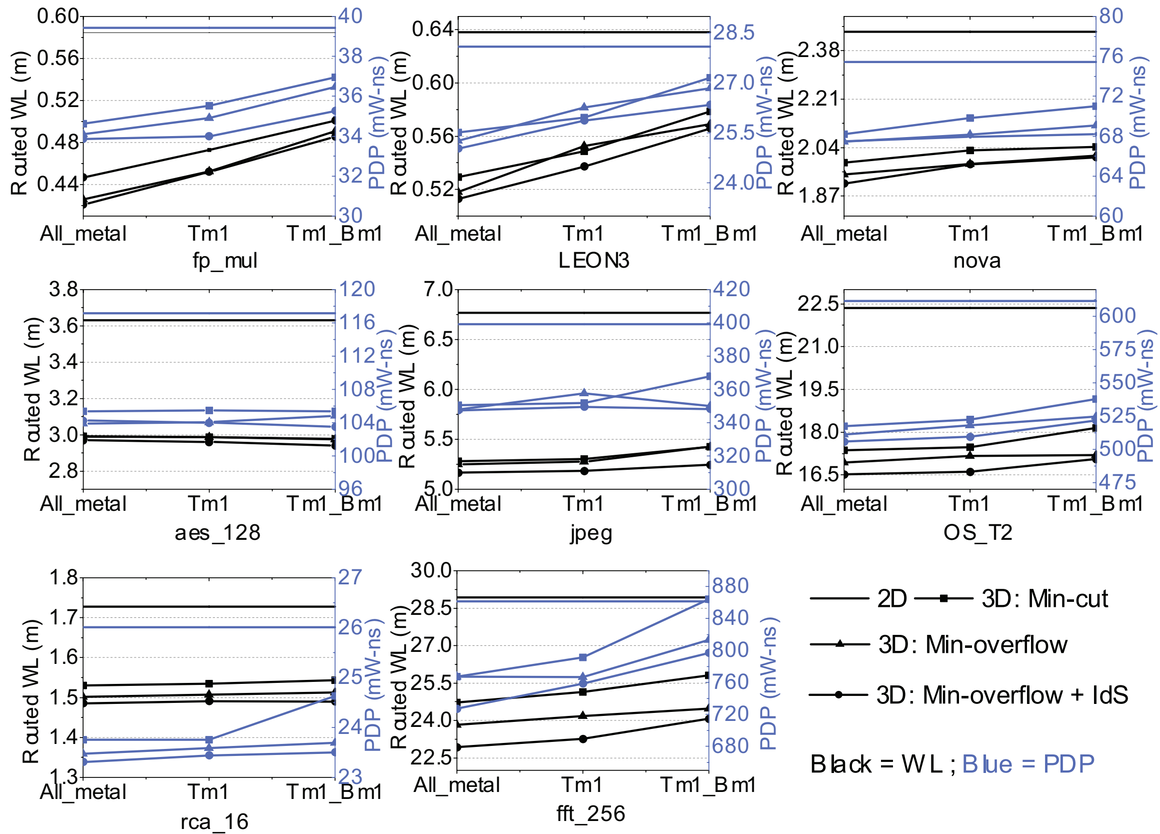


Figure 58: The impact of reducing the metal layer count. "Tm1" ("Bm1") stands for one metal layer removed from the top (bottom) tier.

4.1.5.5 Application to Face-to-Face Bonding

So far, the proposed approach has been applied to monolithic 3D ICs only. However, this approach is general and is applicable to any 3D technology where the via size is so small that the placement need not be aware of them. This section now discusses how this methodology applies to face-to-face (F2F) technology, which has a different stack-up than the face-to-back style discussed so far. The placement engine itself need not change. This is because a design can be placed as if it was face-to-back, and then the mask of the top tier is mirrored with the center of the die as the axis of symmetry. Modifications to the min-overflow algorithm and router-based via insertion steps is now discussed.

For the min-overflow partitioner, F2F without IdS is identical to the monolithic 3D partitioner without IdS. With IdS, only a few changes need to be made. First, the supply in

the F2F layer depends on the top metal layer of *both* tiers, not just the top tier. Therefore, Equation (32) is computed for both tiers separately, and the number of F2F via blockages is the maximum of the two. Next, to calculate the 2D supply reduction, Equation (35) is applied to each tier independently.

For router-based F2F insertion, consider the modified technology LEF file as shown in Figure 54(a). To represent face-to-face, the order of the metal layers of the top die simply need to be reversed. The stack-up will now be $M_{1_1}, \cdots, M_{N_1}$, $M_{N_0}, \cdots, M_{1_0}$. Note that no additional modifications are made to the macro LEF file. In addition, no routing blockages are placed over cells, as F2F vias do not occupy silicon space. Finally, while tracing the routing topology, the F2F landing pads are created on the top metal layers of each tier. Each tier can then be routed, and the mask of the top tier will be mirrored before fabrication. Sample MIV and F2F vias after insertion are shown in Figure 59.

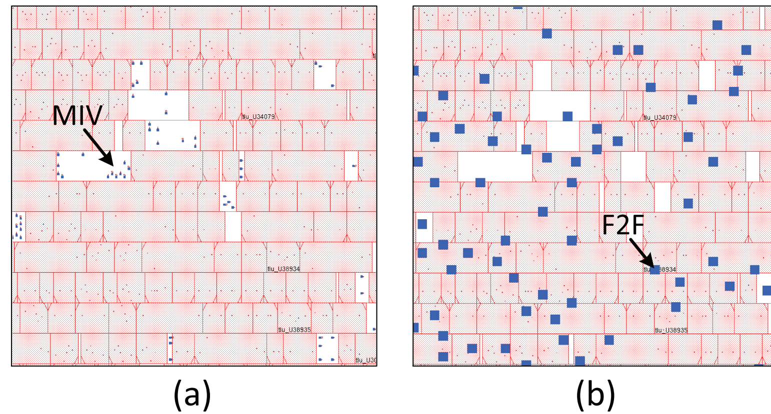


Figure 59: (a) Monolithic 3D integration, and (b) Face-to-face 3D integration. MIVs are limited to whitespace, while F2F vias are not.

F2F vias are assumed to have a width of $0.5\mu m$, a resistance of 0.1Ω, and a capacitance of $0.2fF$. The “Tm1” case is assumed, and the routed WL and PDP for the min-cut and min-overflow (with and without IdS) are tabulated in Table 24. Although the min-overflow partitioner without IdS gives an average WL reduction of 2%, the PDP actually goes up slightly. This is due to overestimation of the available F2F supply, and the more accurate partitioner with IdS corrects this issue.

Table 24: The impact of routability-driven partitioning for face-to-face designs.

Circuit	Min-cut			Min-overflow (w/o IdS)			Min-overflow (with IdS)		
	WL (m)	PDP (mW-ns)	#F2F ($\times 10^3$)	WL (m)	PDP (mW-ns)	#F2F ($\times 10^3$)	WL (m)	PDP (mW-ns)	#F2F ($\times 10^3$)
mul_64	0.49	35.87	5.29	0.47	35.67	6.94	0.46	35.45	6.44
LEON3	0.59	27.29	5.32	0.58	27.92	6.36	0.58	27.06	5.90
nova	2.09	73.57	20.21	2.03	73.89	25.58	2.02	71.30	23.82
rca_16	1.53	23.75	19.34	1.50	23.68	24.36	1.47	23.38	21.97
aes_128	3.01	111.72	52.81	3.05	112.38	60.33	2.97	108.36	62.20
jpeg	5.37	351.63	52.58	5.30	348.76	68.12	5.19	344.97	60.00
OS_T2	17.78	533.25	115.73	17.48	530.55	153.96	17.02	521.85	130.31
fft_256	25.31	762.76	145.04	24.35	757.78	204.96	23.43	735.29	168.82
Geo-Mean	3.01	113.39	29.09	2.95	113.46	37.08	2.90	110.94	33.62
Norm.	1.00	1.00	1.00	0.98	1.00	1.27	0.96	0.97	1.15

4.1.5.6 Overall Comparisons

The WL and PDP numbers of 2D, and the monolithic 3D and face-to-face designs obtained after partitioning with IdS are now compared. The results are tabulated in Table 25. From this table, M3D offers up to a 25.6% WL benefit and 16.6% PDP benefit. On average, M3D offers 19.9% and 11.8% WL and PDP benefit, respectively. In contrast, F2F offers up to 23.8% WL benefit and 14.6% PDP benefit. On average, 18.2% and 10.1% WL and PDP benefit is seen, respectively.

Table 25: Overall Comparisons

Circuit	2D		3D – MIV		3D – F2F	
	WL (m)	PDP (mW-ns)	WL (m)	PDP (mW-ns)	WL (m)	PDP (mW-ns)
mul_64	0.584	39.432	0.452	33.119	0.468	35.454
LEON3	0.638	28.088	0.537	25.863	0.582	27.060
nova	2.447	75.420	1.982	67.947	2.028	71.308
rca_16	1.727	26.010	1.491	23.443	1.474	23.384
aes_128	3.632	117.148	2.961	103.978	2.979	108.365
jpeg	6.769	399.339	5.183	349.531	5.193	344.972
OS_T2	22.352	611.400	16.615	509.550	17.024	521.850
fft_256	28.922	861.750	23.263	758.793	23.436	735.297
Geo-Mean	3.547	123.338	2.842	108.476	2.901	110.941
Norm.	1.000	1.000	0.801	0.880	0.818	0.899

In general, F2F has slightly worse numbers than monolithic 3D. This is because of the larger vias sizes (necessitated by die-alignment) and the fact that connecting two gates in 3D requires a stacked via through both tiers. F2F also has other issues not considered here, such as the requirement of being in a regular array, through-silicon-vias required for I/O connections to the chip, and the non-availability of flip-chip style packaging.

4.1.6 Comparison with Existing 3D Placers

The proposed placer is compared against two existing techniques, which were primarily developed for *TSV-based 3D placement*. The first technique is 3D-Craft [12], which performs true 3D placement, and the other is the partition-then-place approach [28]. No comparison is made against another TSV-specific 3D placer [21], because the binary is not publicly available. In addition, [21] only presents absolute 3D WL numbers without providing *any 2D* baseline number. It is therefore unclear how much of the improvement comes from their 2D engine, and how much from their 3D specific approach. Since the proposed 3D approach can easily incorporate any 2D engine, any engine specific gains in [21] will also carry over.

4.1.6.1 Comparison with 3D-Craft [12]

Only the binary version of this tool is available, and it does not support a target density driven mode. The cells are preset to always be placed with a target density of 1, or *without any whitespace in between them*. Such a placement solution will not have any space for router-driven MIV insertion, and hence is inherently *not routable*. For this reason, only the 3D half-perimeter wirelength (HPWL) is compared in this section. In addition, the binary provided is not capable of handling pre-placed hard macros such as memory. Therefore, in this subsection, only the pure-logic designs are compared.

Both the proposed placer and 3D-Craft are run with the number of dies set to one to give a 2D placement. Next, both placers are run with the number of dies set to two, which gives a 3D placement. Only the improvement in HPWL when going to 3D is compared.

The proposed placer is run with a target density of 1 to match the preset setting of 3D-Craft. 3D-Craft also has a via weight parameter in the cost function (as it is TSV-based), which controls the number of 3D vias. This is set to 0 to make the cost function purely 3D HPWL driven. The results of both placers are tabulated in Table 26.

Table 26: Comparison between 3D-Craft and Our Placer

Circuit	Our HPWL (m)			3D-Craft HPWL (m)		
	2D	3D	3D/2D	2D	3D	3D/2D
mul_64	0.39	0.30	0.77	0.34	0.27	0.79
rca_16	1.15	0.92	0.79	1.22	0.97	0.80
aes_128	2.61	1.93	0.74	2.52	1.87	0.74
jpeg	4.96	3.70	0.74	5.09	3.78	0.74
fft_256	18.95	13.63	0.72	19.57	13.31	0.68
Geo-Mean	2.56	1.93	0.75	2.54	1.90	0.75
Norm.	1.00	1.00	1.00	0.99	0.99	1.00

From this table, both placement approaches produce comparable wirelength improvements when going to 3D. Since the proposed placer takes some steps to minimize the MIV count such as min-cut partitioning, the MIV counts are not compared. The benefit of the proposed approach comes not just from comparable improvements in HPWL, but in the fact that any 2D placer can be easily modified and coupled with our partitioner to give high quality results.

4.1.6.2 Comparison with Partition-then-Place [28]

This technique of 3D placement first performs partitioning, and then simultaneous 2D placement of all the tiers while minimizing 3D HPWL. During placement, it looks at all gates in the 3D space, but does not move gates between tiers. Therefore, the initial partition solution is very important, as it greatly affects solution quality. The same KraftWerk engine is used for both types of placement, so they have identical 2D numbers. The utilization of each circuit is set to 70%, and both placement solutions are taken through router-based MIV insertion to obtain routed WL. To generate initial partitions for the partition-then-place approach, [11] is modified to give any target cutsize between min-cut and max-cut.

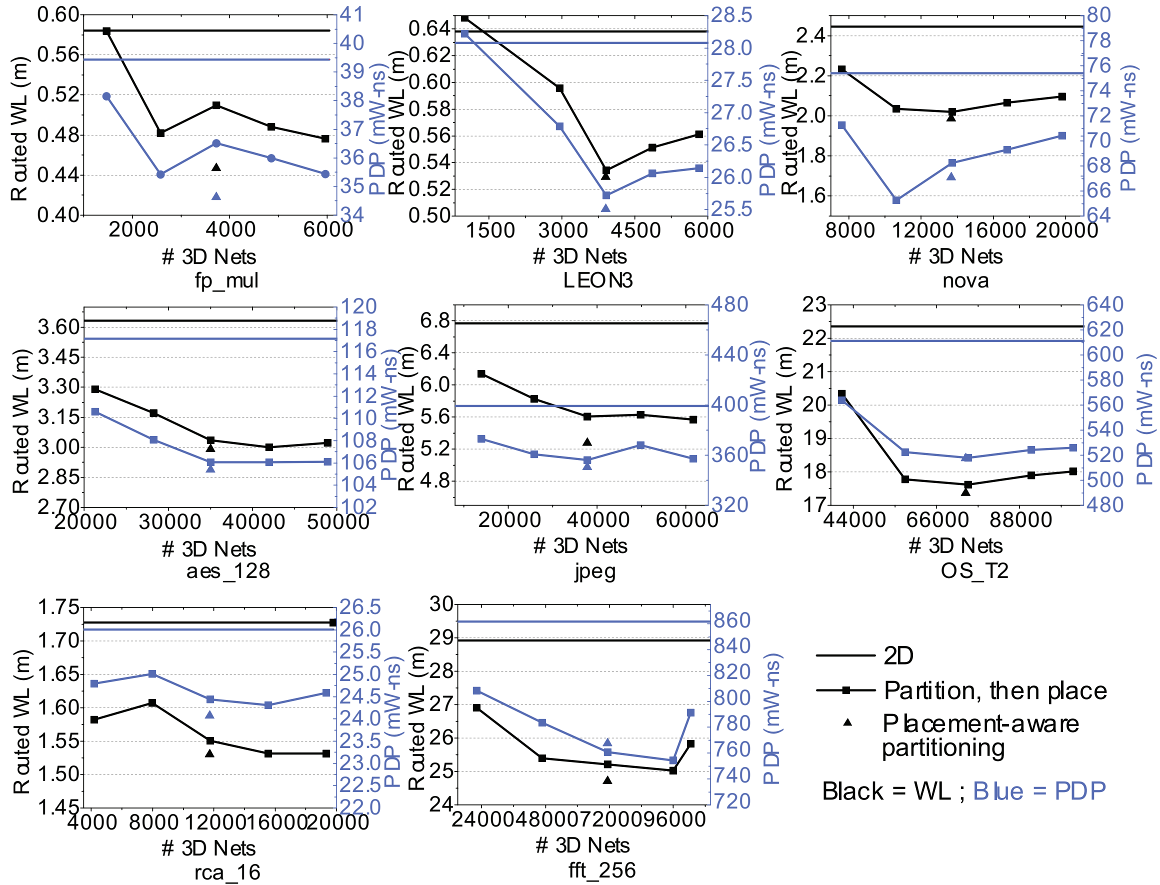


Figure 60: Comparison of 2D, partition-then-place, and placement-aware partitioning methods.

First, the placement-aware partitioning approach is run, and the number of nets used is computed. Partitions are generated starting from this cutsize, in increments of $\pm 5\%$ of the number of nets. The wirelength and PDP for all approaches are plotted in Figure 60. From these graphs, it is clear that choosing an appropriate cutsize is very important to the solution quality. In addition, the proposed approach gives the best wirelength, without the need to sweep the cutsize.

4.2 Monolithic 3D IC Design With Commercial 2D IC Tools

The previous section has described how to modify an academic 2D placer to obtain M3D designs. However, this technique has several limitations. Academic placers usually target wirelength as the objective function, and not timing, which is more critical. In addition, the

techniques in Section 4.1 do not consider timing optimization, while real M3D designs need to be timing closed. Finally, commercial engines include state-of-the art power optimization techniques such as v_t swaps for gates not on the critical path. For a fair comparison with commercial-quality 2D results, M3D needs these optimizations as well. Therefore, this section presents a methodology utilize commercial 2D engines, along with all state of the art optimization steps, to obtain M3D results. The OpenSPARC T2 [52] core is used as a case study throughout this section.

4.2.1 CAD Methodology

This section discusses how the techniques presented in Section 4.1 can be modified to use commercial 2D engines instead of academic ones.

4.2.1.1 Overall Methodology

The overall design flow is shown in Figure 61. First, in order to utilize the 2D tool to handle all the standard cells in a reduced footprint, several technology files are scaled, and this process will be described in detail in Subsection 4.2.1.2. Next, memory handling requires several steps such as memory scaling, memory placement and memory flattening, which will be described in detail in Subsection 4.2.1.3. Once this is done, the commercial 2D engine (Cadence Encounter) can be run on this "shrunk 2D" design (described in Subsection 4.2.1.4). This result is then split into multiple tiers to obtain a DRC-clean sign-off design as described in Subsection 4.2.1.5, and finally timing and power analysis is performed as before.

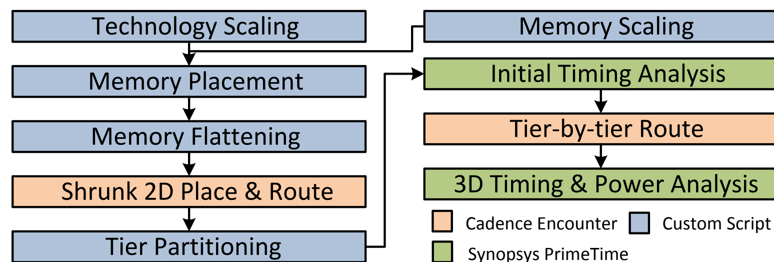


Figure 61: The overall CAD methodology flow used in this paper.

4.2.1.2 Scaling Technology Files

The goal of this step is twofold. The commercial 2D tool first needs to be tricked into placing all the gates in half the footprint area, and it also needs to be able to extract the wire parasitics such that the shrunk 2D design reflects the final geometries in a 3D design. Note that this subsection assumes a gate-only design, and handling memory will be introduced in Subsection 4.2.1.3.

Placing all the gates into half the area can be achieved by shrinking the area of each standard cell by 50%. The width, height and the location of all the pins within the cell are scaled by $1/\sqrt{2}$ (0.707). In addition, the chip width and height are scaled by 0.707 to reduce the 2D footprint area by half. This will also be the footprint of each tier in the final M3D design. Note that since the x and y axis equations in an analytical placer are linear, scaling all the dimensions by 0.707 will simply make the cell locations 0.707 of what they used to be in the 2D placement solution. This leads to a theoretical HPWL improvement of 29.3%.

Next, in order to make the routing in the shrunk 2D accurately represent the routing in monolithic 3D, both the metal width and pitch of each metal layer is shrunk by 0.707. Since the chip width and height are also shrunk by the same amount, the total routing track length does not change between 2D and shrunk 2D. The total track length will also be the same in 3D, and hence this method gives a good estimate of wire length. Note that the wire RC per unit length *is not changed*, even though the wire width is smaller. Therefore, the extracted RC values from the tool does not reflect the geometry of shrunk 2D, but that of a M3D wire of equivalent length using the original metal geometries.

4.2.1.3 Handling Memory Macros

While standard cells can be handled by shrinking their footprint, this is not the case for memory. This is because standard cells can be moved by the placer, while memory is pre-placed. Since no standard cell can be placed in the location where a memory is pre-placed,

simply shrinking the memory is not an option. A pre-placed memory can be thought of as a combination of its pins, which serve as anchors for standard cell placement, and a placement blockage over its footprint, which prevents cells from being placed over it. Each component is described separately.

In order to isolate the memory pin portion, the footprint of the memory is shrunk to the minimum size possible (that of a filler cell). However, the relative locations of its pins are not scaled. This is shown in Figure 62. This will lead to memory pins that are placed outside the memory footprint. These pins will be in the same location they would have been if the memory was its original size. Therefore, from a memory pin perspective, the pre-placed memory in both tiers can simply be shrunk down as described, and fixed in the shrunk 2D footprint.

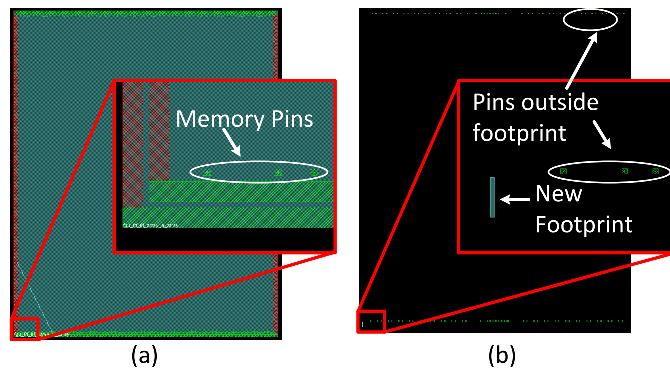


Figure 62: Isolating the memory pins by shrinking the memory footprint. (a) Initial memory footprint, and (b) Memory footprint reduced to size of filler cell.

Handling the placement blockage portion of the memory is similar to what was described in Figure 50. Those regions that have two memories overlapping cannot contain cells in any tier, and hence will become full placement blockages in the shrunk 2D footprint. Those regions that have only one memory can contain cells in the tier where the memory is not placed. In the shrunk 2D design, the maximum placement density of these regions needs to be reduced to reflect this fact. This can be achieved by using partial placement blockages. For example, if the target density of the final 3D design is 70%, then the maximum placement density of the partial placement blockages is set to 35%.

4.2.1.4 Shrunk 2D Place and Route

The shrunk technology and standard cell libraries are fed along with the memory related pins and blockages into Cadence Encounter. This commercial 2D IC tool is then used to run through *all* the design stages such as placement, post-placement optimization, CTS, routing, and post-route optimization. Unlike conventional 3D flows, this approach avoids the problem of tier-by-tier timing optimization. The advantage of this is that the tool can see the entire 3D path, and will insert the minimum buffers required to meet timing.

4.2.1.5 Obtaining a 3D Design

Once the shrunk 2D place and route is done, the cells and memories are expanded back to their original areas. This directly corresponds to results from modified 2D academic placers, and the existing partitioning approaches can be applied to this result. A snapshot of this entire process of obtaining a 3D design using shrunk 2D is shown in Figure 63.

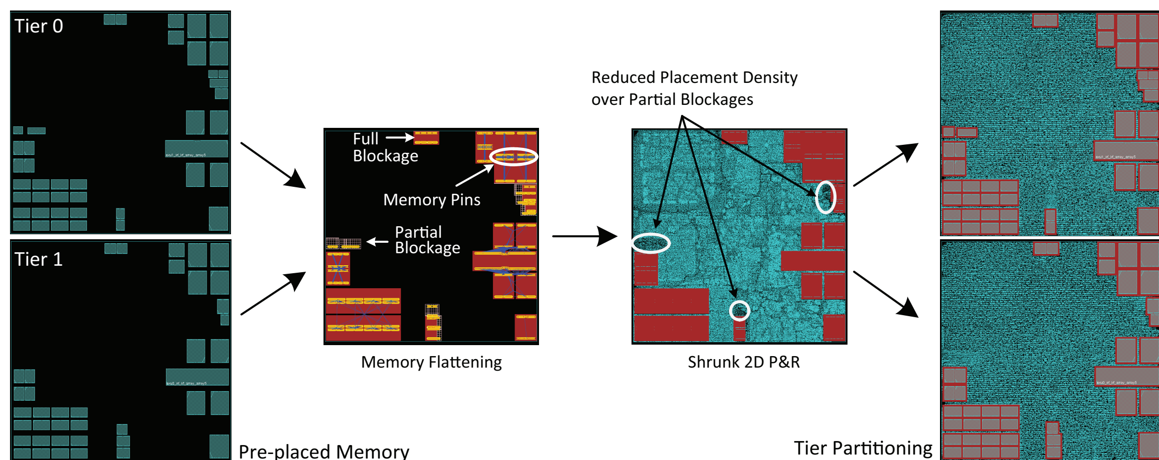


Figure 63: Pre-placed memory is flattened to get a shrunk 2D footprint, on which 2D P&R is performed. This is then partitioned to get a monolithic 3D solution.

In addition to splitting the logic, the commercial flow enables the building of a clock tree in the shrunk 2D design. The conventional approach for 3D ICs (using commercial tools) is to create one separate clock tree per tier, and tie them together using a single MIV. However, the OpenSPARC T2 core has several clock gates built into the RTL. So,

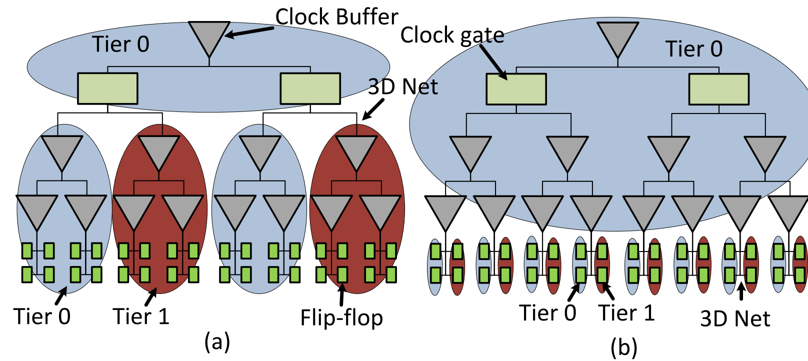


Figure 64: Two different types of 3D CTS possible (a) One clock tree per tier for each gating group (source-level), and (b) The entire backbone is fixed onto tier 0 (leaf-level).

to use the conventional approach, all the clock gating cells are fixed onto tier 0 (as shown in Figure 64(a)), and one clock tree per tier is constructed for each gating group. This is termed source-level CTS, as MIVs are inserted close to the clock source. This approach does not use the clock tree from shrunk 2D at all, so if this approach is to be used, no clock tree is constructed in shrunk 2D, and instead a fixed clock uncertainty is set during optimization.

This section proposes a new CTS methodology that will help reduce the clock power. Since MIVs are very small, it can be assumed that any number of them can be inserted. In this case, the existing CTS result of shrunk 2D can be reused. This clock tree contains several levels of logic as shown in Figure 64(b). During the logic splitting process, the entire clock backbone (clock buffers and clock gates) is fixed onto tier 0. Only the leaf-level flip-flops are free to be partitioned to maintain area balance. Therefore, MIVs will be inserted following all leaf clock buffers that drive flip-flops in both tiers. This approach is termed leaf-level CTS, and an example of this approach for the OpenSPARC T2 core is shown in Figure 65.

Next, the same gate-level MIV insertion scheme can be used. However, for certain nets, the router is bound to insert multiple MIVs. Since existing 3D tool flows use tier-by-tier optimization, timing constraints need to be derived for each tier. In each tier, MIVs are defined as I/O ports, and the timing constraints are captured as input/output delays.

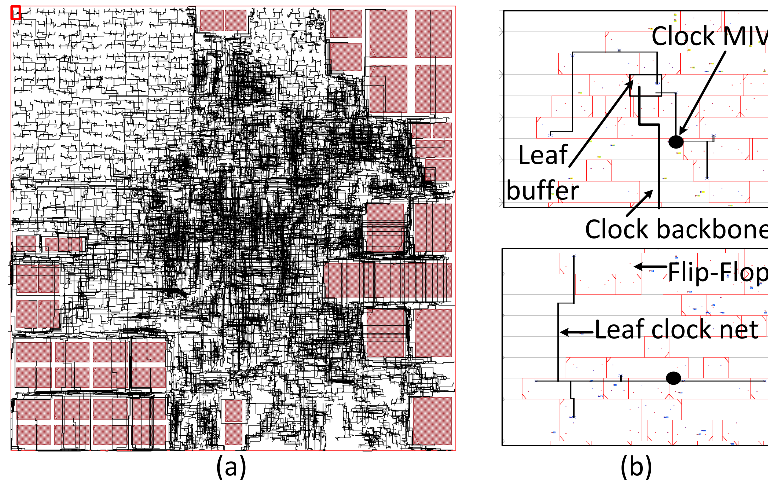


Figure 65: The proposed CTS methodology (a) The clock backbone in tier 0, and (b) Zoom-in shot of leaf-level flip-flops in both tiers connected to a leaf clock buffer in tier 0.

However, if a single net contains multiple MIVs, then it becomes very difficult to capture multiple input/output delays on a single net, as such conditions do not arise in 2D ICs (which current tools are designed for). Therefore, multiple MIV insertion is converted to single MIV insertion by picking the best MIV (in terms of HPWL) from those inserted, and re-routing the net. This could potentially increase the wirelength, but is unavoidable for conventional 3D flows. In the proposed flow, since the optimization is performed in the shrunk 2D design and not tier-by-tier, multiple MIV insertion can be used, which will reduce wirelength and power. Routing topologies for single and multiple MIV insertion for a given net are shown in Figure 66. Once the 3D design is obtained, timing and power analysis can be performed as usual.

4.2.2 Power Benefit Study

The OpenSPARC T2 core is chosen as a case study, and implemented in a 28nm technology library. The power benefit that monolithic 3D ICs offer when compared to a commercial quality sign-off 2D design is investigated. All the numbers presented in this section are for timing closed designs, with a frequency of $1Ghz$. This is the maximum frequency that the 2D version could be design with while using a high-effort timing-driven flow in Cadence

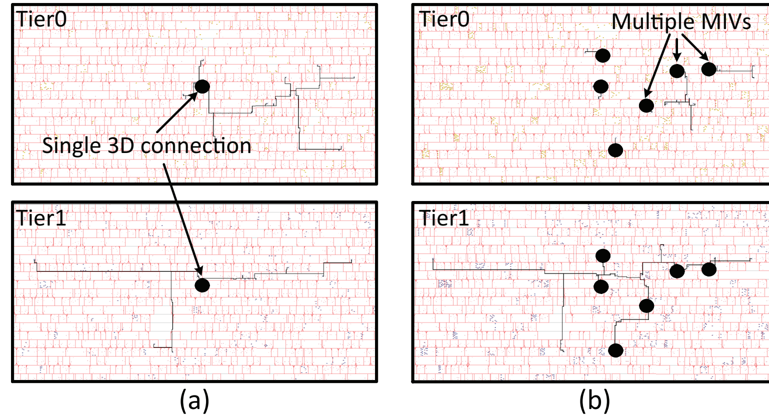


Figure 66: Two types of MIV insertion for a 3D net (a) Single, (b) Multiple

Encounter. The footprint area of the monolithic 3D IC design is exactly half that of the 2D design, and therefore, all 3D designs presented here have zero total silicon area overhead when compared to 2D.

The MIV diameter is assumed to be $100nm$, and its resistance and capacitance are assumed to be 2Ω and $0.1fF$ respectively. Comparisons with face-to-face integration are also provided, and the F2F via diameter, resistance and capacitance are assumed to be $500nm$, 0.5Ω and $0.2fF$ respectively. All required scripts are implemented in C/C++, Python and Tcl.

4.2.2.1 Single vs. Multiple MIV Insertion

The power benefit offered by using multiple MIVs (or F2F vias) for each 3D net is first investigated. A summary of results for both single and multiple MIV insertion is tabulated in Table 27.

From this table, it is observed that using multiple vias offers 8.4% and 10.04% wire-length reduction, for M3D and F2F respectively. In addition, the number of 3D vias double. This means that each net is, on average, using approximately two MIV/F2F vias. This wire-length reduction does not reduce leakage power, but it does reduce some cell power. The biggest reduction is in net power, which reduces by 3.81% and 4.53% for M3D and F2F, which translates to 2.25% and 2.66% total power reduction, respectively.

Table 27: Comparison of single vs. multiple MIV/F2F insertion. Power values are reported in mW, and wirelength in meter.

	Monolithic 3D			Face-to-face		
	Single	Multiple	Diff(%)	Single	Multiple	Diff(%)
Total WL	15.61	14.29	-8.43	15.44	13.89	-10.05
#MIV/F2F	106k	235k	+120.44	106k	202k	+89.72
Total Pwr	**534.10**	**522.10**	**-2.25**	**538.30**	**524.00**	**-2.66**
Cell Pwr	126.90	126.10	-0.63	127.30	126.40	-0.71
Net Pwr	293.90	282.70	-3.81	297.80	284.30	-4.53
Lkg Pwr	113.30	113.30	0.00	113.30	113.30	0.00

Table 28: Comparison of two different types of 3D CTS. Power values are reported in mW, and wirelength in meter.

	Monolithic 3D			Face-to-face		
	Source-level	Leaf-level	Diff (%)	Source-level	Leaf-level	Diff (%)
#MIV/F2F	871	11,376	+1.2k	871	11,376	+1.2k
Skew (ps)	197.42	103.00	-47.83	172.90	117.07	-32.29
Clock Pwr	**68.40**	**48.00**	**-29.82**	**69.00**	**48.50**	**-29.71**
Tier0 WL	0.55	0.62	+11.89	0.53	0.62	+16.61
Tier1 WL	0.48	0.19	-60.50	0.48	0.17	-64.85
Total WL	1.03	0.80	-21.67	1.01	0.79	-21.91
#Tier0 Buf	14,610	21,687	+48.44	14,958	21,687	+44.99
#Tier1 Buf	12,444	0	-100	12,691	0	-100
#Total Buf	27,054	21,687	-19.84	27,649	21,687	-21.56

4.2.2.2 CTS: Source-level vs. Leaf-level

This section discusses the power benefit that the proposed CTS methodology (leaf-level) offers over existing 3D techniques (source-level). A summary of results is tabulated in Table 28. Clearly, leaf-level CTS offers huge reductions in clock skew, as well as a 29.82% reduction in the clock tree power. There are 871 clock-gating related cells in the design, which is why source-level CTS uses that number of MIV/F2F vias. In addition, leaf-level uses far more 3D vias, which helps reduce the clock power.

These power reduction numbers can be explained on the basis of per-tier wirelength and buffer count. Leaf-level CTS uses far more buffers and has a longer WL on tier 0, which is the tier with the clock-backbone. On the other hand, the number of buffers is zero in

tier 1 and the WL is much smaller. In comparison, source-level has a more balanced clock WL and buffer count between the tiers, but this comes at the cost of an increase in the total clock WL and buffer count.

4.2.2.3 Overall Comparisons: 2D vs. 3D

Using the techniques that give the best power reduction (i.e. multiple MIV insertion and leaf-level CTS), M3D and F2F is compared with a 2D IC designed using Cadence Encounter. A summary of results is tabulated in Table 29. From this table, shrunk 2D reduces the wirelength by 27.05% compared to 2D. This is very close to the 29.3% HPWL bound predicted in Section 4.2.1. The improvement number goes down for both M3D and F2F, which is to be expected. In addition, M3D has slightly higher WL compared to F2F because the MIVs are limited to whitespace, while F2F vias are not. Next, the 3D implementations reduce the buffer count by 22.3%, which translates to a 8.03% reduction in total gate count. Since MIV and F2F designs are obtained by simply splitting the shrunk 2D design, all three have the same gate counts. The reduced wirelength and gate count lead to a total power reduction of 15.57% and 15.27% for M3D and F2F respectively. Finally, F2F has a higher power consumption than M3D even though it has lower WL, which is due to increased parasitics of F2F vias. Also, both M3D and F2F power numbers are quite close to the shrunk 2D numbers, which shows that the shrunk 2D design is a very good estimate of M3D and other fine-grained 3D technologies.

The total power is divided into cell, net, and leakage power. The cell power reduces at a number roughly equal to the total gate count reduction. The net power reduces roughly proportional to wirelength, and finally, the leakage reduction is slightly larger than cell count reduction due to smaller buffer sizes. The total power can also be split up by lumping the internal, net and leakage power of certain classes of gates/memory together. This is also tabulated in Table 29. It is observed that the flip-flop clock pin power and register power are virtually unchanged in 3D. The biggest savings in power come from combinational

Table 29: Overall comparisons between 2D and different 3D implementation styles. Power numbers are in mW.

	Enc. 2D	Shrunk 2D		Monolithic 3D		Face-to-face	
Total WL(m)	17.96	13.10	(-27.0%)	14.29	(-20.4%)	13.89	(-22.6%)
# MIV/F2F	-	-		235,394		235,394	
# Buffers	164,917	128,098	(-22.3%)	128,098	(-22.3%)	128,098	(-22.3%)
#Tot. Gates	458,824	421,959	(-8.0%)	421,959	(-8.0%)	421,959	(-8.0%)
Total Pwr	**618.40**	**514.40**	**(-16.8%)**	**522.10**	**(-15.5%)**	**524.00**	**(-15.2%)**
Cell Pwr	135.60	126.80	(-6.4%)	126.10	(-7.0%)	126.40	(-6.7%)
Net Pwr	356.30	274.30	(-23.0%)	282.70	(-20.6%)	284.30	(-20.2%)
Leak. Pwr	126.50	113.30	(-10.4%)	113.30	(-10.4%)	113.30	(-10.4%)
Mem. Pwr	49.00	45.10	(-7.9%)	45.10	(-7.9%)	45.00	(-8.1%)
Comb. Pwr	385.10	300.00	(-22.1%)	305.30	(-20.7%)	306.80	(-20.3%)
Clk Tr. Pwr	62.50	46.90	(-24.9%)	48.00	(-23.2%)	48.50	(-22.4%)
FF Clk Pwr	9.70	9.90	(+2.0%)	9.60	(-1.0%)	9.70	(0.0%)
Reg. Pwr	112.10	112.50	(+0.3%)	114.00	(+1.6%)	114.00	(+1.6%)

logic (20.72% savings), and from the clock tree (23.20% savings). These also exists some memory power savings due to reduction in the output net length that the memory drives.

4.2.2.4 Impact of Dual-Vt Gates

All the results discussed so far have used only the regular V_t standard cell library for both 2D and 3D designs. However, it is known that converting cells on non-critical paths to a high V_t flavor can help reduce leakage power. In this section, dual V_t designs (DVT) are implemented, and their power benefit versus single V_t designs (SVT) is evaluated. For both 2D and 3D (shrunk 2D), Encounter is used to perform leakage optimization during the P&R flow. In addition, leakage optimizations are performed in PrimeTime using a script similar to [19], and the results are tabulated in Table 30.

It is observed that dual V_t M3D designs reduce the total power of 2D designs by 16.08%. This is a slightly better improvement number than the SVT case alone. This is due to the fact that there are more paths that become non-critical in 3D. The F2F improvement numbers are also better than the SVT case. Therefore, the 3D power benefit not only carries over to dual-V_t designs, it actually improves.

Table 30: Dual-Vt comparisons between 2D and different 3D implementation styles. Power is in mW.

	Enc. 2D	Monolithic 3D		Face-to-face	
Total WL(m)	17.94	14.29	(-20.33%)	13.89	(-22.59%)
#MIV/F2F	-	235,394		202,593	
Total Pwr	**572.10**	**480.10**	**(-16.08%)**	**482.20**	**(-15.71%)**
Cell Pwr	131.80	123.00	(-6.68%)	123.30	(-6.45%)
Net Pwr	356.60	282.70	(-20.72%)	284.30	(-20.27%)
Leak. Pwr	83.60	74.40	(-11.00%)	74.60	(-10.77%)
Mem. Pwr	48.80	45.10	(-7.58%)	45.00	(-7.79%)
Comb. Pwr	361.60	283.00	(-21.74%)	284.30	(-21.38%)
Clk Tree Pwr	62.50	48.00	(-23.20%)	48.50	(-22.40%)
FF Clk Pin Pwr	9.10	9.20	(+1.10%)	9.20	(+1.10%)
Reg. Pwr	90.00	94.90	(+5.44%)	94.80	(+5.33%)

4.3 IR-drop Aware Partitioning for Monolithic 3D ICs

The previous two sections have presented techniques to design gate-level monolithic 3D ICs with either academic or commercial 2D engines. Partitioning techniques such as min-cut and min-overflow were also presented. Although sign-off quality designs can be obtained, real design issues such as power delivery and IR-drop was not considered. In three dimensional integration, power delivery to the tier farther away from the package is a problem [38]. This is especially true in monolithic 3D as the vias are very small and hence more resistive than TSVs. The power thus has to traverse the tier closer to the package first, and then pass through a highly resistive stack before it can reach the farther tier. This leads to significant IR-drop in the farther tier. One solution to this problem is moving power hungry cells close to the package. However, in a conventional package, this causes thermal issues, as the majority of the heat is conducted from the heatsink, which is close to the tier farther away from the package. In fact, several thermal optimization works exist that try to solve the temperature issue by moving power hungry cells and modules closer to the heatsink [14]. However, this usually worsens the IR-drop problem, which most works do not consider. Only a handful of works co-optimize thermal and IR-drop in 3D ICs [38]. The approach usually taken to improve IR-drop is to strengthen the power delivery network

(PDN). This has other consequences such as increasing the signal wirelength, total power of chip, and so on. This section presents a partitioning technique that can reduce IR-drop, while also reducing the PDN resource demand.

4.3.1 Motivation and Objectives

In a conventional package, moving power-hungry cells closer to the package usually alleviates the IR-drop problem, but increases the temperature. However, in a mobile package, heat is conducted away from both sides of the chip in equal proportions [1]. Using the simple resistive equivalent circuit of Figure 67, it is demonstrated that the temperature increase is much less of a problem in a mobile package. Note that the resistance values are for illustrative purposes only. The absolute thermal resistance in the mobile package has also been increased to represent the fact that each side conducts heat poorer than a full heat sink. Two partitioning cases are considered – one where the tiers are equally balanced in power, and the other where the tier close to package has 70% of the chip's power.

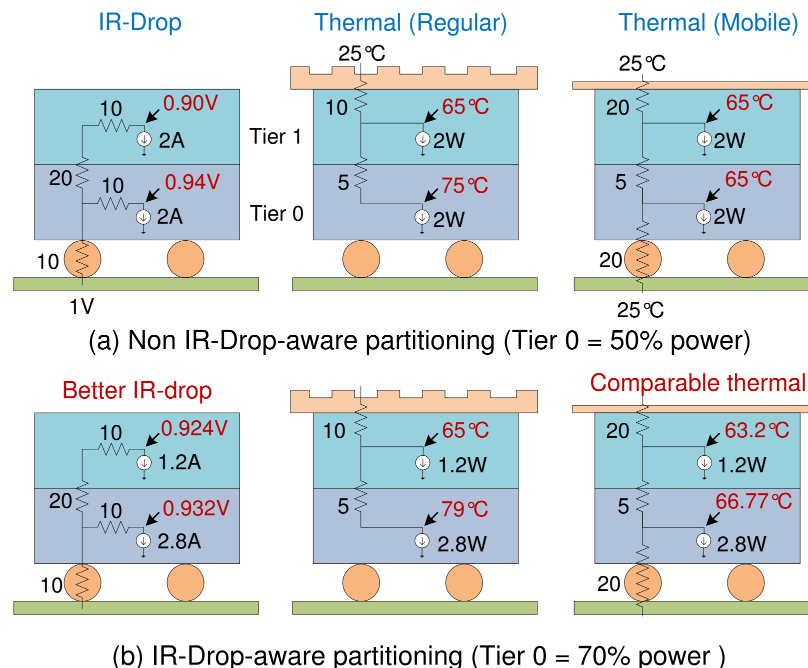


Figure 67: Resistive equivalent circuits for IR-drop and thermal in a conventional and mobile package. Moving high power cells to the tier close to package helps alleviate IR-drop. In a mobile package, the temperature increase is much smaller than in a conventional package. Resistance is in $m\Omega$, and thermal resistance in $^{\circ}C/W$.

It is observed that the IR-drop in the non-optimized partition is quite severe in the farther tier, and that the optimized partition can help reduce the IR-drop by 25%. Next, for the conventional package, moving power close to the package and away from the heat sink leads to a temperature increase of $4°C$. In a mobile package, however, heat is conducted away from both the top and bottom of the chip in roughly equal proportions (details are given in Section 4.3.2.3). In such a scenario, the temperature increases only by $1.7°C$, while still maintaining the same IR-drop benefit.

In addition, it has been demonstrated [57], that increasing the PDN in the farther tier (tier 1) has a significant impact on solution quality. This is because the PDN on the top-metal interferes with MIV insertion, which leads to sub-optimal MIV locations, and this increases the wirelength and degrades solution quality. Therefore, IR-drop aware partitioning will also help reduce the PDN burden on tier 1, thereby improving design quality. Thus, the objective of this section is to obtain a gate-level partition such that the tier closer to the package has more power than the tier farther away from the package, *without degrading solution quality*.

4.3.2 Design and Analysis Flow

An overview of the proposed design flow is shown in Figure 68. "Shrunk2D" design is first performed on the netlist as in the previous section. An initial power analysis is performed on this design to get power numbers for each standard cell. These are kept constant during the partitioning process. Next, this design is partitioned (described in Subsection 4.3.2.1) such that a given power target is met (e.g. 70% power in tier 0, 30% power in tier 1). This solution is legalized, and a PDN is designed for each tier (described in Subsection 4.3.2.2). MIV planning is performed, with a similar flow as before. After obtaining 3D power numbers, accurate 3D IR-drop analysis (Subsection 4.3.2.2) and thermal analysis (Subsection 4.3.2.3) is performed.

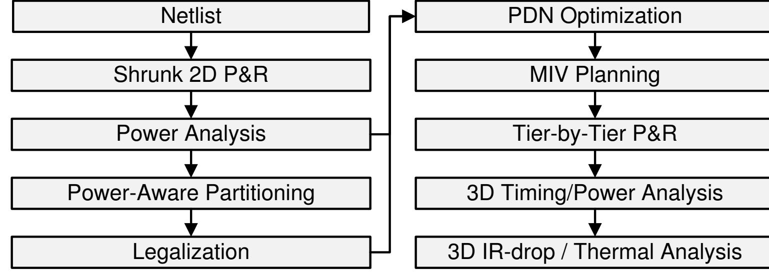


Figure 68: The design flow used for IR-drop-aware partitioning.

4.3.2.1 *IR-drop-aware tier Partitioning*

This subsection describes how placement-aware-partitioning is modified such that the end result meets a certain power target for each tier. In the original partitioning technique, the first step is to create a random, area-balanced partition. A heuristic that generates an initial partition that already satisfies the power targets is proposed in Algorithm 4.

Algorithm 4: Power-aware initial solution generation.

Input: Power targets of each tier `target(`$t0$`)`, `target(`$t1$`)`

Output: An area-balanced solution that meets the targets

1 `areaBalance()`;
2 $tier_{max} \leftarrow$ `max( power(`$t0$`),power(`$t1$`))`;
3 $tier_{min} \leftarrow$ `min( power(`$t0$`),power(`$t1$`))`;
4 $unbalance \leftarrow 0$;
5 **while** `power(`$tier_{max}$`)` $<$ `target(`$tier_{max}$`)` **do**
6   c_{max} = max. power cell from $tier_{min}$;
7   c_{min} = min. power cell from $tier_{max}$;
8   **if** `power(`c_{min}`)` $\geq$ `power(`c_{max}`)` **then** `break`;
9   **if** $ubnalance == 0$ **then**
10   swap c_{max} and c_{min};
11         $unbalance += $ `area(`c_{min}`) - area(`c_{max}`)`;
12   **else if** $unbalance > 0$ **then**
13   move c_{max} to $tier_{min}$;
14         $unbalance -= $ `area(`c_{max}`)`;
15   **else**
16   move c_{min} to $tier_{max}$;
17         $unbalance += $ `area(`c_{min}`)`;
18   **end**
19 **end**

The first step, *areaBalance*, creates a random, area-balanced partition as before (line 1). Next, tiers that have the larger and smaller power targets (lines 2–3) are identified. The next step is to move power from the tier with the smaller power target to the tier with the larger power target without hurting area balance. The cell with maximum power from the tier with smaller power target (c_{max}), and the cell with minimum power from the tier with the larger power target (c_{min}) are identified (lines 6–7). If all cells had equal area, these two could simply be swapped, and this process repeated until the power target was achieved. However, since cells have unequal area, the area unbalance is tracked using an $unbalance$ variable. In essence, one of the two chosen cells is only moved if the area balance target is met (lines 12–17). Cell swaps are terminated if c_{min} has more power than c_{max}, as no further power optimization is possible (line 8).

With this initial solution, the objective is to perform a min-cut as before, without harming the target power distributions. In addition to the area balance condition of the min-cut, a power unbalance condition is defined. If moving a cell from one tier to another makes the power distribution deviate from the target distribution by more than a couple of percent, then that move is illegal. Essentially, a global min-cut subject to *both* area balance and power distribution targets is performed.

4.3.2.2 PDN Design and Analysis

An overview of the PDN structure used is shown in Figure 69(a). First, the power is fed from the C4 bumps to a power-mesh on the tier closer to the package (tier 0). This power mesh consists of thick stripes on the top metal layer, and thinner stripes on an intermediate metal layer. These thinner stripes also have a finer pitch than the top metal layer (Figure 69(b)). This is representative of PDN design for mobile chips [1]. This mesh then connects to local cell rails that feed power to standard cells.

The PDN structure of the tier farther away from the package (tier 1) is quite similar to tier 0, except that it cannot receive power from C4 bumps directly. Instead, MIV arrays

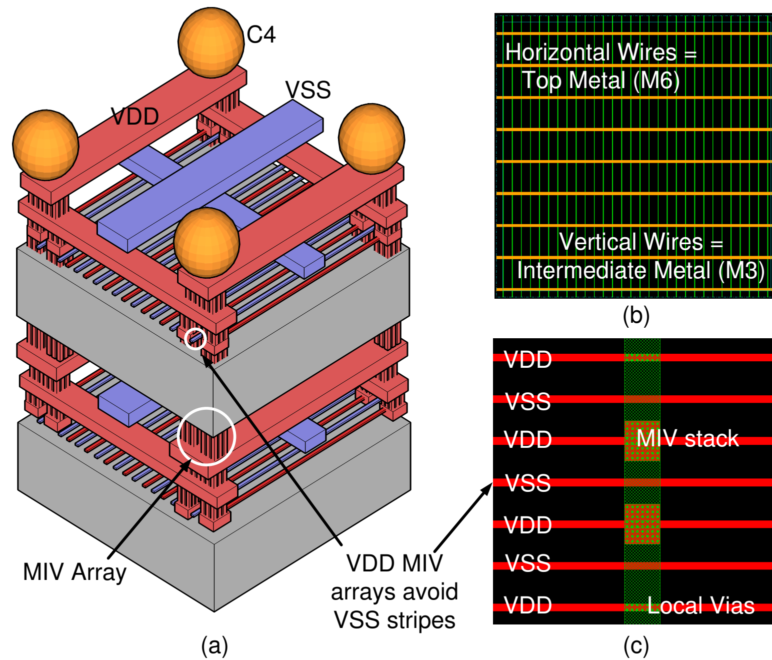


Figure 69: (a) A PDN structure in monolithic 3D. Red wires represent VDD and blue wires represent VSS, (b) The power mesh showing the top and intermediate metal layers, (c) Zoom-in shot of PDN MIV arrays showing only the intermediate mesh layer and local cell rails.

connect the C4 bumps to the PDN mesh on tier 1. While adding these MIV arrays, care must be taken to not short VDD arrays with the thin VSS cell rails. This is achieved by providing a break in the array, as shown in Figures 69(a)&(c).

In order to perform 3D IR-drop analysis, an interconnect technology file that contains all the metal layers and their associated resistivity is created. This is then fed to Cadence Techgen to generate an extraction techfile that can be used for IR-drop analysis. Once the design is completed, two flavors of standard cells are defined, with rails on different metal layers (similar to MIV planning). This is fed along with the power numbers and the extraction techfile to Cadence VoltageStorm to get 3D IR-drop numbers.

4.3.2.3 Thermal Analysis

The structure of a mobile package is shown in Figure 70 [1]. The thickness and thermal properties of the various materials used are tabulated in Table 31. The structure of the chip (excluding package) is taken from [56].

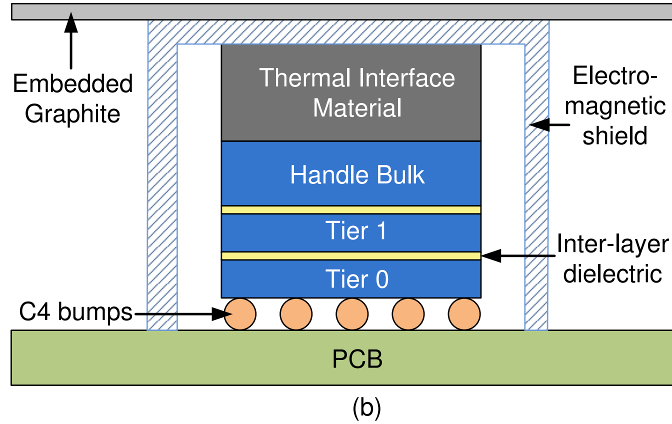


Figure 70: A structure of a mobile package in 3D VLSI [1].

Table 31: Material properties used in a mobile package.

Layer	Thickness (μm)	Thermal cond. (W/mK)	
		Vertical	Lateral
PCB	1200	4.5	60
tier Active	0.1	141	141
Inter-tier ILD	0.1	1.38	1.38
Handle Bulk	75	141	141
TIM	650	5	5
EMI Shield	250	120	120
Graphite Sheet	25	4.5	500

It is observed that the embedded graphite on the top of the chip, as well as the PCB at the bottom have much higher thermal conductivities in the lateral direction than in the vertical direction. This is because graphite is composed of layers of graphene sheets, each of which is highly conductive, and there is very little inter-sheet thermal conduction. Similarly, the majority of the heat conduction in a PCB is through the lateral conduction of the metal planes present in it. There is limited inter-plane heat conduction. Therefore, both act as heat spreaders. Although the PCB has a lower conductivity than graphite, it is thicker and also closer to the chip. Therefore, heat is conducted away in roughly equal proportions from both sides of the chip.

In order to perform thermal analysis, each layer of the 3D structure is meshed into grids of size $20\mu m \times 20\mu m$. The thermal resistance of each tile is computed based on the material within it, and set up as a thermal resistor. In addition, if this tile is in one of the

active layers, then the power in that tile is set up as a current sink in that tile. Boundary conditions are set up as voltage sources at room temperature ($27°C$) on the sides of PCB and the graphite layer, as well as the top of the graphite layer and bottom of the PCB. This entire resistive structure along with voltage sources and current sinks is fed into HSPICE to obtain the node voltages at each mesh tile, which gives the temperature at each and every location.

4.3.3 Experimental Results

4.3.3.1 Experimental Settings

Two benchmarks are chosen, and their statistics are tabulated in Table 32. The first one is a crossbar taken from the OpenSPARC T2 muiti-processor SoC. It is a full 8×8 crossbar that can connect one of 8 cores to any of 8 cache blocks, and vice-versa. The second design is a jpeg encoder taken from the OpenCores benchmark suite.

Table 32: Benchmarks used.

Circuit	Clock (ns)	# Gates	WxH ($\mu m \times \mu m$)		# VDD C4	
			2D	3D	2D	3D
crossbar	1	121,142	600x600	400x400	16	9
jpeg	1.5	255,842	650x650	450x450	16	9

This table shows the clock period at which each design is closed. It also shows the number of gates for 2D and 3D implementations of each design. The gate counts are different as 3D requires fewer buffers for optimization and timing closure. Since no optimization is performed after partitioning, the gate count remains the same for all 3D implementations. Note that both benchmarks have similar footprints, although the gate counts are very different. This is because jpeg contains a lot of small gates, and is more locally connected, whereas the crossbar contains fewer, but larger gates, and is an interconnect dominated design. A $C4$ bump pitch of $100\mu m$ is assumed, which corresponds to a pitch of $200\mu m$ for each of VDD and VSS.

While designing the power delivery network, the width of M6 and M3 wires are assumed to be $4\mu m$ and $1\mu m$, respectively. Only the pitch of these wires is changed to strengthen or weaken the PDN. In addition, in all experiments, the PDN utilization of M3 tracks is assumed to be roughly half the PDN utilization of M6 tracks. This is because it is an intermediate metal layer, and is also needed for signal routing. The diameter of each MIV is assumed to be $100nm$, with a resistance of 2Ω and a capacitance of $0.1fF$ [33]. As depicted in Figure 69(c), each C4 bump has two sets of MIV arrays that carry power to tier 1. Each MIV array has 56 MIVs arranged in a 8×7 array. A foundry $28nm$ SOI library which has a supply voltage of $0.9V$ is used for design and analysis. The IR-drop target is set to be 5% for each of VDD/VSS so that the IR-drop and ground bounce together are within 10%. This corresponds to a IR-drop target of $45mV$.

4.3.3.2 Baseline Designs

The PDN utilization for a 2D IC is chosen by determining the minimum percentage of metal layers that is required to meet the IR-drop target. Next, 3D ICs are designed assuming the same PDN utilization as 2D to obtain baseline designs. Their statistics are tabulated in Table 33. Note that a smaller reduction in wirelength (WL) in the crossbar leads to a larger total power reduction compared to jpeg. This is because it is interconnect dominated. It is also observed that jpeg has a higher power consumption, and therefore requires more PDN resources. As expected, the 3D design does not meet the IR-drop targets with the same PDN utilization as 2D. This is because of both fewer $C4$ bumps and the fact that tier 1 suffers from higher IR-drop. Finally, because a mobile package has heat conduction on both sides, the temperature increase from 2D to 3D is in the range of only $10°C$, even though the power density doubles in 3D. Reducing the 3D IR-drop to acceptable levels is now explored.

Table 33: Design statistics of baseline 2D and 3D designs.

Circuit	WL (m)		Power (mW)		M6/M3	Drop (mV)		Temp ($^{\circ}C$)	
	2D	3D	2D	3D	PDN%	2D	3D	2D	3D
crossbar	3.68	3.12	137.5	125.8	15/8	45	79	64.2	71.6
jpeg	3.26	2.53	222.9	213.6	30/15	39	73	80.25	92.25

4.3.3.3 PDN Sensitivity Analysis

As discussed in Section 4.3.2, the objective is to partition the design such that tier 0 has more power than tier 1. This will lead to reduced PDN demand, improving solution quality. Now suppose $x\%$ of power is moved from tier 1 to tier 0, and $y\%$ of PDN resources in tier 1 are freed up. The additional $x\%$ of power in tier 0 should require less than $y\%$ additional PDN in tier 0 in order to get a net benefit. In order to validate this assumption, the power consumed in each tier is scaled, and the resulting change in that tier's IR-drop is plotted in Figure 71.

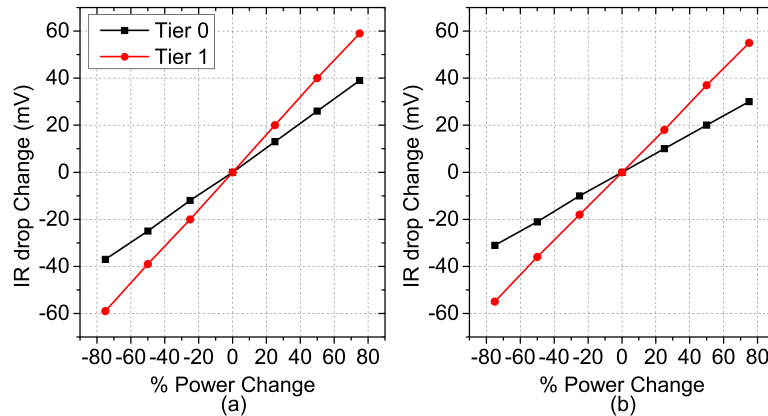


Figure 71: Sensitivity of tier IR-drop to change in tier power for (a) crossbar, and (b) jpeg.

From this figure, it is seen that a transfer of 30% power from tier 1 to tier 0 reduces the tier 1 IR-drop by a much greater margin than the tier 0 IR-drop is increased. For example, removing 30% power from tier 1 in the crossbar benchmark reduces the tier 1 IR-drop by $30mV$. This power is added to tier 0, but the graph shows that this increases the tier 0 IR-drop by only $15mV$. This makes it much easier to fix any remaining IR-drop violations. In addition, the reduced PDN demand will reduce chip power and improve design quality.

4.3.3.4 IR-drop-aware Partitioning Results

This section maintains the same PDN density as the baseline designs, applies the IR-drop-aware partitioning technique, and demonstrates that under the same PDN, significant reduction in IR-drop can be achieved. Different target power distributions are given to the partitioner, starting with 30% power on tier 0 (30/70), and changed in increments of 10% all the way till 70% power on tier 0 (70/30). The resulting statistics of each design is tabulated in Table 34. From this table, the 70/30 and 30/70 targets do not give the required distributions exactly. Therefore, it is concluded that 65% power on one tier is the most power unbalance achievable in these designs. This is reasonable given that the tiers need to be area balanced. It is unlikely that half the cells (w.r.t. area) will consume more than 70% of the power.

Next, it is observed that providing a power target impacts the cutsize of the partitioner. This is because an additional power constraint is added on top of the existing area balance constraints. The MIV planner inserts more than one MIV per 3D net when appropriate, so its count is more than the cutsize. The cutsize increase is also reflected in the MIV count. In general, since MIVs are small, more of them can be tolerated. This is observed in the fact that, except for a few outliers, the WL increase is quite small. This leads to only a minor increase in the total power of the design.

However, the impact on the IR-drop is dramatic. Up to a 24.66% reduction in the maximum IR-drop of the chip can be achieved, with a thermal impact of $< 1°C$. The 30/70 and 40/60 partitions are also tabulated as they are the conventional “thermal-aware” partitions, where power is moved towards the heat sink. Although the temperature reduces in these partitions, the IR-drop increases significantly. The IR-drop benefit is also plotted in Figure 72, which clearly shows the IR-drop reduction by clever partitioning.

Table 34: The impact of IR-drop-aware partitioning. The PDN utilization is kept the same as the baseline designs.

Power (T0/T1%)		Cutsize	#MIV	WL (m)	Total Power (mW)	IR Drop (mV) Tier0 / Tier1	Temp. ($^{\circ}C$) Tier0 / Tier1
Target	Actual						
crossbar							
Baseline	47.1 / 52.9	17,868 -	30,772 -	3.124 -	125.8 -	50 / 79 -	71.65 / 70.84 -
30 / 70	33.5 / 66.5	23,419 (+31.1%)	34,764 (+12.9%)	3.60 (+15.3%)	128.1 (+1.83%)	40 / 105 (+32.9%)	71.59 / 71.61 (-0.06%)
40 / 60	40.2 / 59.8	18,242 (+2.1%)	31,552 (+2.5%)	3.14 (+0.54%)	125.9 (+0.08%)	40 / 87 (+10.1%)	71.16 / 70.81 (-0.68%)
50 / 50	50.9 / 49.1	17,968 (+0.6%)	30,836 (+0.2%)	3.13 (+0.32%)	125.9 (+0.08%)	58 / 82 (+3.80%)	71.62 / 70.76 (-0.04%)
60 / 40	59.3 / 40.7	15,840 (-11.4%)	26,993 (-12.3%)	3.16 (+1.03%)	126.1 (+0.24%)	68 / 67 (-13.9%)	72.32 / 70.81 (+0.94%)
70 / 30	**65.8 / 34.2**	**21,282 (+19.1%)**	**30,313 (-1.5%)**	**3.12 (-0.11%)**	**125.9 (+0.08%)**	**75 / 56 (-5.06%)**	**72.5 / 70.69 (+1.19%)**
jpeg							
Baseline	44.6 / 55.4	34,834 -	41,122 -	2.53 -	213.6 -	41 / 73 -	92.25 / 91.63 -
30 / 70	35.2 / 64.8	51,772 (+48.6%)	56,982 (+38.6%)	2.58 (+1.89%)	214.1 (+0.23%)	29 / 85 (+16.4%)	91.83 / 91.87 (-0.41%)
40 / 60	39.9 / 60.1	41,528 (+19.2%)	47,666 (+15.9%)	2.56 (+1.17%)	213.8 (+0.09%)	37 / 79 (+8.22%)	91.93 / 91.71 (+0.07%)
50 / 50	49.9 / 50.1	34,527 (-0.9%)	40,452 (-1.6%)	2.53 (+0.21%)	213.7 (+0.05%)	46 / 66 (-9.59%)	92.07 / 91.58 (+0.15%)
60 / 40	**58.1 / 41.9**	**35,540 (+2.0%)**	**40,695 (-1.1%)**	**2.53 (+0.11%)**	**213.6 (+0.00%)**	**53 / 55 (-24.6%)**	**92.53 / 91.48 (+0.50%)**
70 / 30	64.8 / 35.2	58,859 (+68.9%)	62,798 (+52.7%)	2.58 (+2.05%)	214.4 (+0.37%)	57 / 45 (-21.9%)	92.69 / 91.62 (+0.17%)

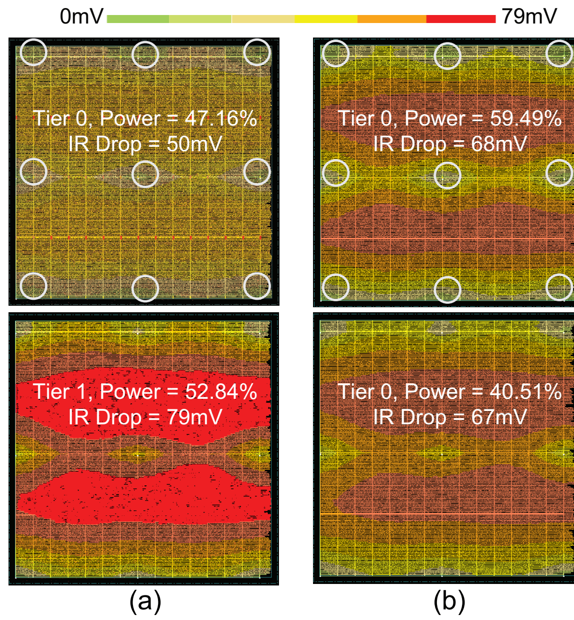


Figure 72: IR-drop maps for crossbar benchmark. (a) baseline, (b) our IR-drop-aware partition, where tier 0 has 60% of the chip power.

4.3.3.5 PDN Resource Optimization

The previous section demonstrated that under the same PDN utilization, significant IR-drop reduction can be achieved. However, the IR-drop numbers for many designs were significantly over the budget, and needs to be fixed. In this section, explores optimizing the PDN of each tier such that the IR-drop target $(45mV)$ is met. To do this, the results of the previous section are taken, and the PDN resources required to meet the IR-drop target is estimated. The 3D IC is redesigned with this estimate, and if it still does not meet the target, the estimate is revised. This is repeated until the target is met. For the sake of simplicity, the ratio between the utilization of M6 and M3 is kept the same. In addition, the maximum utilization of M6 is set to 75%. If a design still does not meet the IR-drop target with 75% M6 utilization, IR-drop is not optimized further. The results of these optimizations are tabulated in Table 35.

Table 35: The impact of PDN optimization such that the IR-drop falls within the $45mV$ target.

Pow. Dist. (T0/T1)	PDN M6/M3 % Tier 0	PDN M6/M3 % Tier 1	PDN M6/M3 % Change	#MIV	WL (m)	Power (mW) Total	IR Drop T0/T1 (mV)	Temp. ($^{\circ}C$) Tier0/Tier1
crossbar								
Baseline	24 / 12	68 / 36	-	27,265 -	3.25 -	128.1 -	37 / 41	72.34 / 71.68 -
30 / 70	15 / 8	75 / 40	0.00%	31,540 (+15.68%)	3.71 (+13.95%)	131 (+2.26%)	36 / 52	72.66 / 72.67 (+0.46%)
40 / 60	15 / 8	75 / 40	0.00%	26,594 (-2.46%)	3.31 (+1.80%)	129.2 (+0.86%)	40 / 45	72.32 / 72.02 (-0.03%)
50 / 50	15 / 8	60 / 32	-8.33%	27,675 (+1.50%)	3.23 (-0.63%)	127.6 (-0.39%)	44 / 44	72.28 / 71.45 (-0.08%)
60 / 40	30 / 16	45 / 24	-16.67%	25,526 (-6.38%)	3.22 (-1.11%)	127.1 (-0.78%)	41 / 40	72.64 / 71.14 (+0.41%)
70 / 30	**38 / 20**	**30 / 16**	**-25.00%**	**29,828 (+9.40%)**	**3.19 (-1.85%)**	**126.9 (-0.94%)**	**40 / 39**	**72.91 / 71.02 (+0.79%)**
jpeg								
Baseline	30 / 15	75 / 38	-	38,264 -	2.68 -	215.5 -	38 / 48	92.71 / 92.24 -
30 / 70	22 / 11	75 / 38	-7.14%	54,772 (+43.14%)	2.81 (+4.70%)	217.2 (+0.79%)	37 / 49	92.76 / 92.79 (+0.09%)
40 / 60	22 / 11	75 / 38	-7.14%	45,278 (+18.33%)	2.72 (+1.34%)	216 (+0.23%)	46 / 52	92.71 / 92.43 (+0.00%)
50 / 50	38 / 19	75 / 38	7.14%	37,829 (-1.14%)	2.69 (+0.25%)	215.7 (+0.09%)	39 / 43	92.82 / 92.23 (+0.12%)
60 / 40	**45 / 18**	**45 / 18**	**-14.29%**	**39,979 (+4.48%)**	**2.58 (-4.03%)**	**214.4 (-0.51%)**	**41 / 46**	**92.6 / 91.63 (-0.12%)**
70 / 30	45 / 18	30 / 15	-28.57%	62,809 (+64.15%)	2.58 (-3.74%)	214.7 (-0.37%)	45 / 45	92.57 / 91.64 (-0.15%)

From this table, it is observed that the PDN utilization can be reduced by up to 28.57% from the baseline, and still meet IR-drop targets. In some cases, the baseline is not able to meet the IR-drop target even with PDN optimization, as the initial IR-drop is too severe. This reduction in the PDN utilization, especially in tier 1, frees up additional resources for signal routing and MIV insertion. This gives up to a 4% reduction in the total WL of the design. This helps reduce the chip power, which limits the temperature increase.

The PDN as well as the IR-drop for both the baseline and the 70/30 implementation of the crossbar is plotted in Figure 73. It is clearly seen that there is a huge reduction in the PDN utilization, while the same IR-drop is maintained.

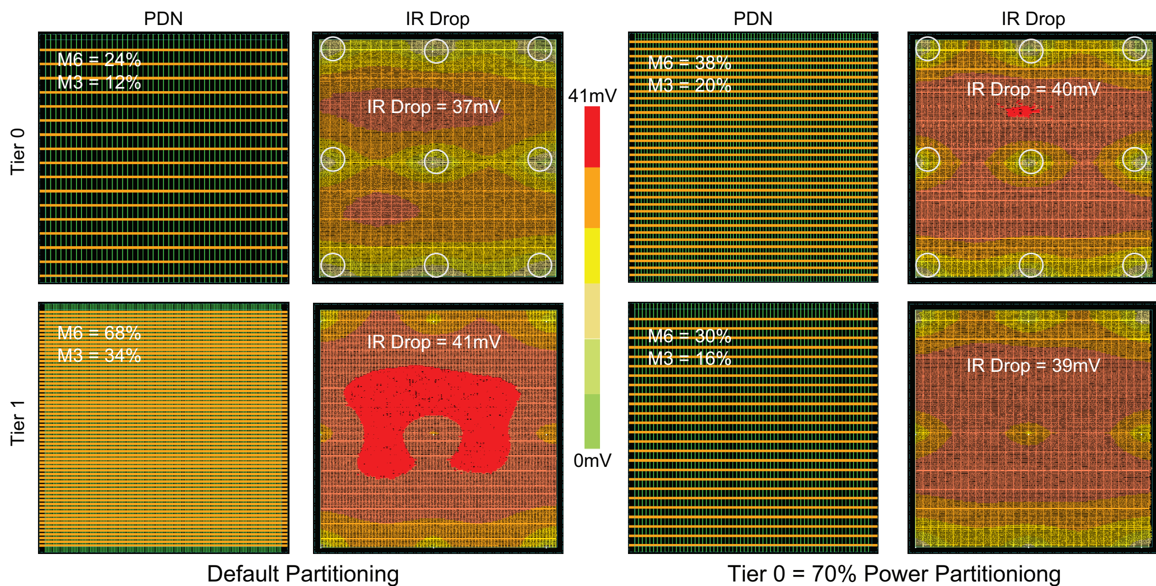


Figure 73: The impact of PDN optimization on the crossbar benchmark. IR-drop aware partitioning is able to achieve the same IR-drop target as the baseline partition while using significantly fewer PDN resources.

The temperature maps for various partition solutions of the crossbar in are shown in Figure 74. Even though an additional 30% power is moved to the bottom tier, the power reduction coupled with the mobile package results in a temperature increase of less than $< 1°C$.

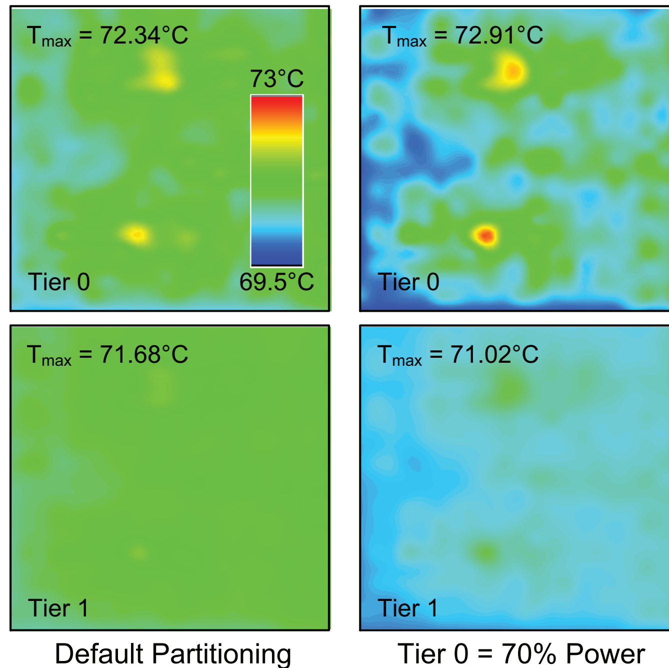


Figure 74: The impact of changing the target power of the bottom tier on the temperature of the crossbar benchmark. Even if the bottom tier has 70% of the chip power, the temperature increase is $< 1°C$.

4.4 Summary

This chapter fist demonstrated that modified 2D placement coupled with a placement-aware partitioning step is sufficient to produce high quality monolithic 3D IC placement results. A router-based MIV insertion algorithm that makes previously unroutable designs routable was presented. A monolithic 3D demand model was used to build a min-overflow partitioning heuristic, and it was demonstrated that this helps to reduce the routed wirelength. Next, a technique to utilize commercial 2D engines instead of academic ones was presented. This enables gate-level monolithic 3D IC designs to be taken all the way through place, route, CTS, and timing optimization. This chapter finally demonstrates that in mobile applications, power can be moved to the tier closer to the package to reduce IR-drop, while not hurting temperature. An IR-drop-aware partitioner was developed that can reduce the power and IR-drop of a monolithic 3D IC, without increasing the maximum operating temperature of the chip.

CHAPTER V

CONCLUSIONS AND FUTURE DIRECTIONS

As discussed in this dissertation, testability for TSV-based 3D ICs remain one of the last challenges facing their adoption. While TSV-based 3D ICs solve some interconnect issues, they do not fully exploit the flexibility of the third dimension. In addition, it was demonstrated that monolithic 3D ICs offer significant benefits over both 2D ICs and TSV-based 3D ICs. Although this is a longer term technology, and the fabrication process is not yet completely mature, physical design techniques are needed to evaluate the benefits of monolithic 3D. In general, before significant resources can be diverted to ramp up monolithic 3D, studies of their efficacy are necessary. To carry out reasonable and meaningful studies, the following are crucial: (1) Physical design techniques for different design styles of monolithic 3D ICs, (2) An understanding of how the fabrication process affects the potential benefits and how to overcome any potential degradation, and (3) An understanding of real world reliability issues such as thermal, IR-drop, e.t.c. that affect monolithic 3D ICs.

Towards these objectives of overcoming the last hurdles of short term TSV-based 3D ICs and developing tools and techniques for evaluating longer term monolithic 3D ICs, the following projects have been presented in this dissertation.

- Design for Test for TSV-based 3D ICs including scan chain construction techniques, a transition delay fault test architecture, IR-drop studies, and test time estimation during 3D IC partitioning.

- Physical design for block-level monolithic 3D ICs, where a floorplanning framework was presented, and extended to consider inter-tier performance differences arising because of an immature fabrication process.

- Physical design for gate-level monolithic 3D ICs, where placement techniques were developed for monolithic 3D ICs. This was extended to utilize commercial tools for placement, timing optimization and CTS. In addition, IR-drop aware partitioning was presented.

The DfT research carried out in this dissertation addressed some of the testability concerns of TSV-based 3D ICs. However, several more hurdles need to be surmounted before TSV-based 3D IC testing can mature. For example, at-speed of TSVs need to be performed before bonding, and the test architecture presented in this dissertation does not support this. In addition, it is as yet unclear under what conditions pre-bond test will be necessary and cost effective.

The floorplanner presented in this dissertation provides a good framework to design block-level monolithic 3D ICs. It was also demonstrated that tungsten interconnects are preferable to degraded transistors. However, it is unclear how this will change at future nodes, where the interconnect is expected to become more of a bottleneck. Additional research needs to be carried out to determine the most effective technology stackup at future nodes.

Finally, an efficient gate-level framework was presented that provides commercial-quality monolithic 3D IC designs. However, it still relies on tricking 2D tools into designing 3D ICs. There are bound to be inaccuracies introduced due to this abstraction, and future research needs to look into development of true 3D tools.

Finally, although physical design was presented for block and gate-level monolithic 3D ICs, today's industrial SoCs are bound to require a mix of the two. For example, large blocks can be implemented in 3D using the gate-level framework, and these 3D blocks can then be assembled together. Additional physical design tools are needed to develop a mixed block and gate-level flow, and these will undoubtedly lead to better quality monolithic 3D IC designs.

REFERENCES

[1] "Personal communication with industry partner.."

[2] Batude, P., Ernst, T., Arcamone, J., Arndt, G., Coudrain, P., and Gaillardon, P.-E., "3-D Sequential Integration: A Key Enabling Technology for Heterogeneous Co-Integration of New Function With CMOS," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, pp. 714–722, Dec 2012.

[3] Batude, P., Vinet, M., Pouydebasque, A., Le Royer, C., Previtali, B., Tabone, C., Hartmann, J.-M., Sanchez, L., Baud, L., Carron, V., Toffoli, A., Allain, F., Mazzocchi, V., Lafond, D., Thomas, O., Cueto, O., Bouzaida, N., Fleury, D., Amara, A., Deleonibus, S., and Faynot, O., "Advances in 3D CMOS sequential integration," in *Proc. IEEE Int. Electron Devices Meeting*, pp. 1–4, Dec 2009.

[4] Bobba, S., Chakraborty, A., Thomas, O., Batude, P., Ernst, T., Faynot, O., Pan, D., and De Micheli, G., "CELONCEL: Effective design technique for 3-D monolithic integration targeting high performance integrated circuits," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 336–343, Jan 2011.

[5] Brenner, U. and Rohe, A., "An effective congestion-driven placement framework," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, pp. 387–394, April 2003.

[6] C. Cabral, J., Fletcher, B., Rossnagel, S., Hu, C.-K., Baker-ONeal, B., Huang, Q., der Straten, O. V., Nitta, S., Rodbell, K., and Edelstein, D., "Metallization Opportunities and Challenges for Future Back-End-of-the-Line Technology," in *Advanced Metallization Conference*, pp. 136–137, October 2010.

[7] Chen, P.-L., Lin, J.-W., and Chang, T.-Y., "IEEE Standard 1500 Compatible Delay Test Framework," *IEEE Trans. on VLSI Systems*, vol. 17, pp. 1152–1156, Aug 2009.

[8] Chen, Q., Davis, J., Zarkesh-Ha, P., and Meindl, J., "A compact physical via blockage model," *IEEE Trans. on VLSI Systems*, vol. 8, pp. 689–692, Dec 2000.

[9] Choi, D., Kim, C. S., Naveh, D., Chung, S., Warren, A. P., Nuhfer, N. T., Toney, M. F., Coffey, K. R., and Barmak, K., "Electron mean free path of tungsten and the electrical resistivity of epitaxial (110) tungsten films," *Phys. Rev. B*, vol. 86, p. 045432, Jul 2012.

[10] Chu, C. and Wong, Y.-C., "FLUTE: Fast Lookup Table Based Rectilinear Steiner Minimal Tree Algorithm for VLSI Design," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, pp. 70–83, Jan 2008.

[11] CONG, J. and LIM, S. K., "Edge separability-based circuit clustering with application to multilevel circuit partitioning," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, pp. 346–357, March 2004.

[12] CONG, J. and LUO, G., "A multilevel analytical placement for 3D ICs," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 361–366, Jan 2009.

[13] CONG, J., LUO, G., WEI, J., and ZHANG, Y., "Thermal-Aware 3D IC Placement Via Transformation," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 780–785, Jan 2007.

[14] CONG, J., WEI, J., and ZHANG, Y., "A thermal-driven floorplanning algorithm for 3D ICs," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, pp. 306–313, Nov 2004.

[15] DONG, X., ZHAO, J., and XIE, Y., "Fabrication Cost Analysis and Cost-Aware Design Space Exploration for 3-D ICs," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, pp. 1959–1972, Dec 2010.

[16] FIDUCCIA, C. and MATTHEYSES, R., "A Linear-Time Heuristic for Improving Network Partitions," in *Proc. ACM Design Automation Conf.*, pp. 175–181, June 1982.

[17] GOEL, S. K. and MARINISSEN, E. J., "SOC Test Architecture Design for Efficient Utilization of Test Bandwidth," *ACM Trans. on Design Automation of Electronics Systems*, vol. 8, pp. 399–429, Oct. 2003.

[18] GOLSHANI, N., DERAKHSHANDEH, J., ISHIHARA, R., BEENAKKER, C. I. M., ROBERTSON, M., and MORRISON, J., "Monolithic 3D integration of SRAM and image sensor using two layers of single grain silicon," in *IEEE International 3D System Integration Conference*, pp. 1–4, Nov 2010.

[19] GUPTA, P., KAHNG, A., SHARMA, P., and SYLVESTER, D., "Gate-length biasing for runtime-leakage control," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, pp. 1475–1485, Aug 2006.

[20] HE, X., DONG, S., MA, Y., and HONG, X., "Simultaneous buffer and interlayer via planning for 3D floorplanning," in *Proc. Int. Symp. on Quality Electronic Design*, pp. 740–745, March 2009.

[21] HSU, M.-K., CHANG, Y.-W., and BALABANOV, V., "TSV-aware analytical placement for 3D IC designs," in *Proc. ACM Design Automation Conf.*, pp. 664–669, June 2011.

[22] JIANG, L., HUANG, L., and XU, Q., "Test architecture design and optimization for three-dimensional SoCs," in *Proc. Design, Automation and Test in Europe*, pp. 220–225, April 2009.

[23] JIANG, L., XU, Q., CHAKRABARTY, K., and MAK, T., "Layout-driven test-architecture design and optimization for 3D SoCs under pre-bond test-pin-count constraint," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, pp. 191–196, Nov 2009.

[24] JIANG, Z.-W., SU, B.-Y., and CHANG, Y.-W., "Routability-driven analytical placement by net overlapping removal for large-scale mixed-size designs," in *Proc. ACM Design Automation Conf.*, pp. 167–172, June 2008.

[25] JUNG, S.-M., JANG, J., CHO, W., MOON, J., KWAK, K., CHOI, B., HWANG, B., LIM, H., JEONG, J., KIM, J., and KIM, K., "The revolutionary and truly 3-dimensional 25F2 SRAM technology with the smallest S3 (stacked single-crystal Si) cell, 0.16um2, and SSTFT (atacked single-crystal thin film transistor) for ultra high density SRAM," in *IEEE Int. Symposium on VLSI Technology*, pp. 228–229, June 2004.

[26] JUNG, S.-M., LIM, H., KWAK, K., and KIM, K., "A 500-MHz DDR High-Performance 72-Mb 3-D SRAM Fabricated With Laser-Induced Epitaxial c-Si Growth Technology for a Stand-Alone and Embedded Memory Application," *IEEE Trans. on Electron Devices*, vol. 57, pp. 474–481, Feb 2010.

[27] KARKLIN, K., BROZ, J., and MANN, B., "Bond Pad Damage Tutorial," in *IEEE Semiconductor Wafer Test Workshop*, June 2008.

[28] KIM, D. H., ATHIKULWONGSE, K., and LIM, S. K., "A study of Through-Silicon-Via impact on the 3D stacked IC layout," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, pp. 674–680, Nov 2009.

[29] KIM, D. H., TOPALOGLU, R., and LIM, S. K., "Block-level 3D IC design with through-silicon-via planning," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 335–340, Jan 2012.

[30] KIM, M.-C., HU, J., LEE, D.-J., and MARKOV, I., "A SimPLR method for routability-driven placement," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, pp. 67–73, Nov 2011.

[31] KNECHTEL, J., MARKOV, I., and LIENIG, J., "Assembling 2-D Blocks Into 3-D Chips," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, pp. 228–241, Feb 2012.

[32] LEE, H.-H. and CHAKRABARTY, K., "Test Challenges for 3D Integrated Circuits," *IEEE Design and Test of Computers*, vol. 26, pp. 26–35, Sept 2009.

[33] LEE, Y.-J., LIMBRICK, D., and LIM, S. K., "Power benefit study for ultra-high density transistor-level monolithic 3D ICs," in *Proc. ACM Design Automation Conf.*, pp. 1–10, May 2013.

[34] LEE, Y.-J., MORROW, P., and LIM, S. K., "Ultra high density logic designs using transistor-level monolithic 3D integration," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, pp. 539–546, Nov 2012.

[35] LEWIS, D. and LEE, H., "A scan island based design enabling prebond testability in die-stacked microprocessors," in *Proc. IEEE Int. Test Conference*, pp. 1–8, Oct 2007.

[36] LEWIS, D., PANTH, S., ZHAO, X., LIM, S. K., and LEE, H.-H., "Designing 3D test wrappers for pre-bond and post-bond test of 3D embedded cores," in *Proc. IEEE Int. Conf. on Computer Design*, pp. 90–95, Oct 2011.

[37] LI, C., XIE, M., KOH, C.-K., CONG, J., and MADDEN, P., "Routability-Driven Placement and White Space Allocation," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, pp. 858–871, May 2007.

[38] LI, Z., MA, Y., ZHOU, Q., CAI, Y., WANG, Y., HUANG, T., and XIE, Y., "Thermal-aware power network design for IR drop reduction in 3D ICs," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 47–52, Jan 2012.

[39] LIU, C. and LIM, S. K., "A Design Tradeoff Study with Monolithic 3D Integration," in *Proc. Int. Symp. on Quality Electronic Design*, pp. 529–536, March 2012.

[40] LIU, C. and LIM, S. K., "Ultra-high density 3D SRAM cell designs for monolithic 3D integration," in *Proc. IEEE Int. Interconnect Technology Conference*, pp. 1–3, June 2012.

[41] LO, C.-Y., WANG, C.-H., CHENG, K.-L., HUANG, J.-R., WANG, C.-W., WANG, S.-M., and WU, C.-W., "STEAC: A Platform for Automatic SOC Test Integration," *IEEE Trans. on VLSI Systems*, vol. 15, pp. 541–545, May 2007.

[42] LOPEZ, G., *The impact of interconnect process variations and size effects for gigascale integration*. PhD thesis, Georgia Institute of Technology, 2009.

[43] MANN, W., TABER, F., SEITZER, P., and BROZ, J., "The leading edge of production wafer probe test technology," in *Proc. IEEE Int. Test Conference*, pp. 1168–1195, Oct 2004.

[44] MARINISSEN, E., CHI, C.-C., VERBREE, J., and KONIJNENBURG, M., "3D DfT architecture for pre-bond and post-bond testing," in *IEEE International 3D System Integration Conference*, pp. 1–8, Nov 2010.

[45] MARINISSEN, E., IYENGAR, V., and CHAKRABARTY, K., "A set of benchmarks for modular testing of SOCs," in *Proc. IEEE Int. Test Conference*, pp. 519–528, 2002.

[46] MARINISSEN, E., VERBREE, J., and KONIJNENBURG, M., "A structured and scalable test access architecture for TSV-based 3D stacked ICs," in *IEEE VLSI Test Symposium*, pp. 269–274, April 2010.

[47] MARINISSEN, E.J. AND ZORIAN, Y., "Testing 3D chips containing through-silicon vias," in *Proc. IEEE Int. Test Conference*, pp. 1–11, Nov 2009.

[48] Naito, T., Ishida, T., Onoduka, T., Nishigoori, M., Nakayama, T., Ueno, Y., Ishimoto, Y., Suzuki, A., Chung, W., Madurawe, R., Wu, S., Ikeda, S., and Oyamatsu, H., "World's first monolithic 3D-FPGA with TFT SRAM over 90nm 9 layer Cu CMOS," in *IEEE Int. Symposium on VLSI Technology*, pp. 219–220, June 2010.

[49] Noia, B., Chakrabarty, K., Goel, S., Marinissen, E., and Verbree, J., "Test-Architecture Optimization and Test Scheduling for TSV-Based 3-D Stacked ICs," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, pp. 1705–1718, Nov 2011.

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

[51] [Online], "OpenCores Benchmark Suite." http://www.opencores.org/.

[52] [Online], "Oracle OpenSPARC T2."

[53] [Online], "International Technology Roadmap for Semiconductors 2009." http://www.itrs.net/, 2009.

[54] Plombon, J. J., Andideh, E., Dubin, V. M., and Maiz, J., "Influence of phonon, geometry, impurity, and grain size on Copper line resistivity," *Applied Physics Letters*, vol. 89, no. 11, pp. –, 2006.

[55] Rajendran, B., Shenoy, R., Witte, D., Chokshi, N., De Leon, R., Tompa, G., and Fabian, R., "Low Thermal Budget Processing for Sequential 3-D IC Fabrication," *IEEE Trans. on Electron Devices*, vol. 54, pp. 707–714, April 2007.

[56] Samal, S., Panth, S., Samadi, K., Saedi, M., Du, Y., and Lim, S. K., "Fast and accurate thermal modeling and optimization for monolithic 3D ICs," in *Proc. ACM Design Automation Conf.*, pp. 1–6, June 2014.

[57] Samal, S., Samadi, K., Kamal, P., Du, Y., and Lim, S. K., "Full chip impact study of power delivery network designs in monolithic 3D ICs," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, pp. 565–572, Nov 2014.

[58] Spindler, P. and Johannes, F., "Fast and Accurate Routing Demand Estimation for Efficient Routability-driven Placement," in *Proc. Design, Automation and Test in Europe*, pp. 1–6, April 2007.

[59] Spindler, P., Schlichtmann, U., and Johannes, F., "Kraftwerk2: A Fast Force-Directed Quadratic Placement Approach Using an Accurate Net Model," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, pp. 1398–1411, Aug 2008.

[60] STEINHOGL, W., STEINLESBERGER, G., PERRIN, M., SCHEINBACHER, G., SCHINDLER, G., TRAVING, M., and ENGELHARDT, M., "Tungsten interconnects in the nano-scale regime ," *Microelectronic Engineering*, vol. 82, pp. 266 – 272, 2005.

[61] TSAI, M.-C., WANG, T.-C., and HWANG, T., "Through-Silicon Via Planning in 3-D Floorplanning," *IEEE Trans. on VLSI Systems*, vol. 19, pp. 1448–1457, Aug 2011.

[62] VUCUREVICH, T., "The Long Road to 3D Integration: Are we there yet?." Key note speech at the 3D Architecture Conference, 2007.

[63] WU, X., FALKENSTERN, P., CHAKRABARTY, K., and XIE, Y., "Scan-chain Design and Optimization for Three-dimensional Integrated Circuits," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 5, pp. 9:1–9:26, July 2009.

[64] WU, X., ZHAO, W., NAKAMOTO, M., NIMMAGADDA, C., LISK, D., GU, S., RADOJCIC, R., NOWAK, M., and XIE, Y., "Electrical Characterization for Intertier Connections and Timing Analysis for 3-D ICs," *IEEE Trans. on VLSI Systems*, vol. 20, pp. 186–191, Jan 2012.

[65] XU, C., BATUDE, P., VINET, M., MOUIS, M., CASSE, M., SKLENARD, B., COLOMBEAU, B., RAFHAY, Q., TABONE, C., BERTHOZ, J., PREVITALI, B., MAZURIER, J., BRUNET, L., BREVARD, L., KHAJA, F., HARTMANN, J., ALLAIN, F., TOFFOLI, A., KIES, R., LE ROYER, C., MORVAN, S., POUYDEBASQUE, A., GARROS, X., PAKFAR, A., TAVERNIER, C., FAYNOT, O., and POIROUX, T., "Improvements in low temperature ($<$625C) FDSOI devices down to 30nm gate length," in *IEEE Int. Symposium on VLSI Technology, Systems, and Applications*, pp. 1–2, April 2012.

[66] YANG, K., KIM, D. H., and LIM, S. K., "Design quality tradeoff studies for 3D ICs built with nano-scale TSVs and devices," in *Proc. Int. Symp. on Quality Electronic Design*, pp. 740–746, March 2012.

[67] ZHAO, X., LEWIS, D., LEE, H.-H., and LIM, S. K., "Pre-bond testable low-power clock tree design for 3D stacked ICs," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, pp. 184–190, Nov 2009.

PUBLICATIONS

This dissertation is based on and/or related to the works and results presented in the following publications in print:

[1] **Shreepad Panth** and Sung Kyu Lim, "Scan Chain and Power Delivery Network Synthesis for Pre-Bond Test of 3D ICs", in *IEEE VLSI Test Symposium*, pp. 26–31, 2011.

[2] Dean Lewis, **Shreepad Panth**, Xin Zhao, Sung Kyu Lim, and Hsien-Hsin Lee, "Designing 3D Test Wrappers for Pre-bond and Post-bond Test of 3D Embedded Cores", in *IEEE International Conference on Computer Design*, pp. 90–95, 2011.

[3] **Shreepad Panth** and Sung Kyu Lim, "Transition Delay Fault Testing of 3D ICs with IR-Drop Study", in *IEEE VLSI Test Symposium*, pp. 270–275, 2012.

[4] Young-Joon Lee, **Shreepad Panth**, and Sung Kyu Lim, "Enabling High Density Logic Designs for Monolithic 3D ICs", in *SRC TECHCON Conference*, 2012.

[5] Brandon Noia, **Shreepad Panth**, Krishnendu Chakrabarty, and Sung Kyu Lim, "Scan Test of Die Logic in 3D ICs Using TSV Probing", in *IEEE International Test Conference*, pp. 1–8, 2012.

[6] Sergej Deutsch, Krishnendu Chakrabarty, **Shreepad Panth**, and Sung Kyu Lim, "TSV Stress-Aware ATPG for 3D Stacked ICs", in *IEEE Asian Test Symposium*, pp. 31–36, 2012.

[7] Sergej Deutsch, Krishnendu Chakrabarty, **Shreepad Panth**, and Sung Kyu Lim, "TSV Stress-Aware ATPG for 3D Stacked ICs", in *IEEE International Workshop on Testing Three-Dimensional Stacked Integrated Circuits*, 2012.

[8] **Shreepad Panth**, Kambiz Samadi, Yang Du, and Sung Kyu Lim, “High-Density Integration of Functional Modules Using Monolithic 3D-IC Technology”, in *IEEE/ACM Asia South Pacific Design Automation Conference*, pp. 681–686, 2013.

[9] **Shreepad Panth**, Kambiz Samadi, and Sung Kyu Lim, “Test-TSV Estimation During 3D-IC Partitioning”, in *IEEE International 3D Systems Integration Conference*, pp. 1–7, 2013.

[10] **Shreepad Panth**, Kambiz Samadi, Yang Du, and Sung Kyu Lim, “Placement-Driven Partitioning for Congestion Mitigation in Monolithic 3D IC Designs”, in *ACM International Symposium on Physical Design*, pp 47–54, 2014.

[11] **Shreepad Panth**, Kambiz Samadi, Yang Du, and Sung Kyu Lim, “Power-Performance Study of Block-Level Monolithic 3D-ICs Considering Inter-Tier Performance Variations”, in *ACM Design Automation Conference*, pp. 1–6, 2014.

[12] **Shreepad Panth**, Kambiz Samadi, Yang Du, and Sung Kyu Lim, “Design and CAD Methodologies for Low Power Gate-level Monolithic 3D ICs”, in *IEEE International Symposium on Low Power Electronics and Design*, pp. 171–176, 2014.

[13] **Shreepad Panth**, Sandeep Samal, Yun Seop Yu, and Sung Kyu Lim, “Design Challenges and Solutions for Ultra-High-Density Monolithic 3D ICs”, in *IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference*, pp. 1–2, 2014.

[14] **Shreepad Panth**, Sandeep Samal, Yun Seop Yu, and Sung Kyu Lim, “Design Challenges and Solutions for Ultra-High-Density Monolithic 3D ICs”, in *Journal of Information and Communication Convergence Engineering*, Vol. 12, No. 3, pp. 186–192, 2014.

[15] **Shreepad Panth**, Kambiz Samadi, Yang Du, and Sung Kyu Lim, “Tier-Partitioning for Power Delivery vs Cooling Tradeoff in 3D VLSI for Mobile Applications”, in *ACM Design Automation Conference*, 2015, to appear.

[16] **Shreepad Panth**, Kambiz Samadi, Yang Du, and Sung Kyu Lim, “Placement-Driven Partitioning for Congestion Mitigation in Monolithic 3D IC Designs”, in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, to appear.

[17] Brandon Noia, **Shreepad Panth**, Krishnendu Chakrabarty, and Sung Kyu Lim, “Scan Test of Die Logic in 3D ICs Using TSV Probing”, in *IEEE Transactions on Very Large Scale Integration Systems*, to appear.

In addition, the author has completed works unrelated to this dissertation presented in the following publications in print:

[1] Moongon Jung, **Shreepad Panth**, and Sung Kyu Lim, “A Study of TSV Variation Impact on Power Supply Noise”, in *IEEE International Interconnect Technology Conference*, pp. 8–12, 2011.

[2] Dae Hyun Kim, Krit Athikulwongse, Michael B. Healy, Mohammad M. Hossain, Moongon Jung, Ilya Khorosh, Gokul Kumar, Young-Joon Lee, Dean L. Lewis, Tzu-Wei Lin, Chang Liu, **Shreepad Panth**, Mohit Pathak, Minzhen Ren, Guanhao Shen, Taigon Song, Dong Hyuk Woo, Xin Zhao, Joungho Kim, Ho Choi, Gabriel H. Loh, Hsien-Hsin S. Lee, and Sung Kyu Lim, “3D-MAPS: 3D Massively Parallel Processor with Stacked Memory”, in *IEEE International Solid-State Circuits Conference*, pp. 188–190, 2012.

[3] Junghee Lee, Chryostomos Nicopoulos, Hyung Gyu Lee, **Shreepad Panth**, Sung Kyu Lim, and Jongman Kim, “IsoNet: Hardware-Based Job Queue Management for Many-Core Architectures”, in *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 21, No. 6, pp. 1080–1093, 2013.

[4] Sandeep Samal, **Shreepad Panth**, Kambiz Samadi, Mehdi Saeidi, Yang Du, and Sung Kyu Lim, “Fast and Accurate Thermal Modeling and Optimization for Monolithic 3D ICs”, in *ACM Design Automation Conference*, pp. 1–6, 2014.

[5] Ahmet Ceyhan, Moongon Jung, **Shreepad Panth**, Sung Kyu Lim, and Azad Naeemi, "Impact of Size Effects in Local Interconnects for Future Technology Nodes: A Study Based on Full-Chip Layouts", in *IEEE International Interconnect Technology Conference*, pp. 345–348, 2014.

[6] Dae Hyun Kim, Krit Athikulwongse, Michael B. Healy, Mohammad M. Hossain, Moongon Jung, Ilya Khorosh, Gokul Kumar, Young-Joon Lee, Dean L. Lewis, Tzu-Wei Lin, Chang Liu, **Shreepad Panth**, Mohit Pathak, Minzhen Ren, Guanhao Shen, Taigon Song, Dong Hyuk Woo, Xin Zhao, Joungho Kim, Ho Choi, Gabriel H. Loh, Hsien-Hsin S. Lee, and Sung Kyu Lim, "Design and Analysis of 3D-MAPS (3D Massively Parallel Processor with Stacked Memory)", in *IEEE Transactions on Computers*, Vol.64, no.1, pp.112–125, 2015.

[7] Ahmet Ceyhan, Moongon Jung, **Shreepad Panth**, Sung Kyu Lim, and Azad Naeemi, "Evaluating Chip-Level Impact of Cu/low-k Performance Degradation on Circuit Performance at Future Technology Nodes", in *IEEE Transactions on Electron Devices*, to appear.

VITA

Shreepad Panth was born in Pune, India, in 1988. He received his B.E. from Anna University, India, in 2009, in Electrical and Electronics Engineering. He also received an M.S from the school of Electrical and Computer Engineering at Georgia Institute of Technology in 2011, where he is currently a Ph.D. candidate under the supervision of Dr. Sung Kyu Lim. His research interests lie in physical design methodologies for monolithic 3D ICs. He is the author of more than 20 publications in top conferences and journals, and has received the best paper award at ATS'12 and nominations for best paper awards at ISPD'14 and DAC'14.