The Pennsylvania State University
The Graduate School

DESIGN METHODOLOGIES OF THREE-DIMENSIONAL INTEGRATED CIRCUITS (3D ICS)

A Dissertation in
Computer Science and Engineering
by
Qiaosha Zou

© 2015 Qiaosha Zou

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2015

The dissertation of Qiaosha Zou was reviewed and approved* by the following:

Yuan Xie
Professor of Computer Science and Engineering
Dissertation Co-Advisor, Co-Chair of Committee

Mary Jane Irwin
Professor of Computer Science and Engineering
Robert E. Noll Professor
Evan Pugh Professor
Disseration Co-Advisor, Co-Chair of Committee

Vijaykrishnan Narayanan
Professor of Computer Science and Engineering

Donghai Wang
Associate Professor of Mechanical Engineering

Lee Coraor
Associate Professor of Computer Science and Engineering
Chair of Department Graduate Program

*Signatures are on file in the Graduate School.

Abstract

The continuous technology scaling results in the growing delay gap between transistors and interconnects because of the significant increase in parasitics. Moreover, the increased integration density and design complexity exacerbate the interconnect issue from both the rising routing requirements and the prolong wirelength. Recently, the emerging three-dimensional integrated circuits (3D ICs) have been studied intensively as one potential solution towards the future high performance and energy efficient computing systems. Different from previous system-in-package (SiP) designs that stack multiple chips and use wires or bumps for connections, the emerging 3D integration provides finer granularity integration thanks to the vertical interconnects inside chips. In general, 3D ICs provide numerous advantages over traditional 2D IC designs, such as smaller footprint, high bandwidth and short latency interconnects, and the capability of heterogeneous stacking.

Nevertheless, there are several challenges in 3D ICs that need to be solved before this technology is applied in commercial designs with high volume production, such as higher fabrication cost, compromised system reliability, the lack of mature electrical design automation tools, the elevated chip operation temperature, and the insufficient understanding on chip testings. The relatively complicate fabrication process implicates higher cost of 3D ICs compared to 2D counterparts. As the cost is the primary driving force for the new technology adoption, reducing the system cost becomes the one of the primary concerns in 3D designs. On the other hand, the success of this emerging technology should be guaranteed by its functionality correctness. However, a few factors influence the 3D reliability: fabrication limitations, thermal mechanical stresses, interconnect electrical failures, degraded signal integrity, and IR droop in power networks. As a result, these factors should be considered and properly addressed in 3D designs to ensure the chip reliability.

This dissertation proposes novel design methodologies to optimize 3D designs emphasizing the challenges of cost and reliability. In the first part, a cost model is adapted and applied in our analysis framework to evaluate the 3D system cost. Then cost-aware design methods are proposed to reduce the cost from fabrication level to chip design level. The second part of this dissertation handles the chip reliability problems. Three studies are introduced from diverse aspects to manage the interconnect electromigration, thermal mechanical stresses, and the signal integrity issues.

Table of Contents

List of Figures **ix**

List of Tables **xiii**

Acknowledgments **xiv**

Chapter 1
Introduction **1**
1.1 3D Integration Technologies 1
1.2 Existing Physical Design Methodologies of 3D ICs 4
 1.2.1 3D Partitioning and Floorplanning 4
 1.2.2 3D Placement and Routing 5
1.3 Existing System Level Design Methodologies 7
1.4 Opportunities and Challenges in 3D ICs 8
1.5 Contribution and Organization 9

Chapter 2
Cost Analysis Framework with Test Elimination **11**
2.1 Preliminaries and Motivation 13
2.2 Cost Benefit Analysis Framework 16
2.3 Hierarchical Yield Model 19
2.4 Experiment and Cost Analysis 21
 2.4.1 Impacts of Design Target 22
 2.4.2 Impacts of Cost Parameters and Defect Clustering Degree . 24
 2.4.3 System Cost Analysis on Multicore Design 25
2.5 Summary 26

Chapter 3

Metal Layer Reduction for Cost Optimization 27

3.1 Routability and Cost Models . . . 28
  3.1.1 Routability Models . . . 28
  3.1.2 3D Cost Model . . . 30
3.2 Cost Efficient 3D Design Exploration Flow . . . 32
3.3 Experiment Results . . . 35
  3.3.1 Cost Analysis without Optimization . . . 35
  3.3.2 Cost Analysis with Optimization . . . 37
3.4 Summary . . . 39

Chapter 4
Vertical Bandwidth Reconfigurable 3D NoCs Utilizing Redundant TSVs 40
4.1 Preliminaries and Related Work . . . 42
  4.1.1 3D Multicore Processor Designs . . . 42
  4.1.2 TSV Redundancy . . . 43
  4.1.3 Motivation and Related Work . . . 44
4.2 Reconfigurable Vertical Link Design . . . 45
  4.2.1 Basic Router Architecture . . . 45
  4.2.2 Proposed Architecture Modification . . . 46
    4.2.2.1 Switch Allocator Design . . . 47
    4.2.2.2 Crossbar Design . . . 47
    4.2.2.3 Additional Link Arbitrator . . . 47
  4.2.3 Design Issues . . . 48
    4.2.3.1 Router Placement . . . 49
    4.2.3.2 Routing Algorithm and Timing Analysis . . . 50
    4.2.3.3 Flow Control . . . 50
    4.2.3.4 Availability of Spare TSVs . . . 51
4.3 Simulation Result Analysis . . . 51
  4.3.1 Area Evaluation . . . 53
  4.3.2 Performance Evaluation . . . 53
  4.3.3 Layer Sensitivity . . . 56
  4.3.4 Failure Mode Sensitivity . . . 57
4.4 Summary . . . 59

Chapter 5
Electromigration Lifetime Analysis in TSV Arrays 60
5.1 Background and Motivation . . . 61
  5.1.1 Motivation . . . 61

5.1.2 Related Work . . . 63
5.2 TSV Array Current Density and EM Lifetime Simulation Mechanism 64
5.2.1 Power Grid Model . . . 65
5.2.2 Current Density Calculation . . . 68
5.2.3 Array EM Lifetime Calculation . . . 69
5.3 TSV Array EM Lifetime Analysis . . . 71
5.3.1 TSV Array EM Lifetime . . . 71
5.3.2 TSV Filling Material . . . 73
5.3.3 TSV Number and Size in Array . . . 74
5.4 Case Study . . . 76
5.5 Summary . . . 77

Chapter 6
Crosstalk Minimization in TSV Arrays 78
6.1 Preliminaries . . . 79
6.1.1 Crosstalk in TSV Array . . . 79
6.1.2 Related Work . . . 81
6.2 3D LAT Coding Mechanism . . . 82
6.2.1 Preliminaries in 2D NAT Code . . . 82
6.2.2 3D ω-LAT Code . . . 83
6.2.3 LAT Code Optimization . . . 87
6.2.4 Heuristic LAT CODEC Design . . . 89
6.3 Evaluation . . . 90
6.3.1 Interconnect Power Analysis . . . 90
6.3.2 Crosstalk Delay Analysis . . . 93
6.4 Summary . . . 94

Chapter 7
Thermo-Mechanical Stresses Management in 3D ICs 95
7.1 Background and Related Work . . . 95
7.1.1 Analysis of TSV Thermal Stress . . . 96
7.1.2 TSV Lateral Thermal Blockage Effect . . . 97
7.1.3 3D Thermal Cycling Effect . . . 97
7.2 Thermomechanical Stress-Aware 3D Design Methodology . . . 98
7.2.1 The Heuristic Floorplan Flow . . . 98
7.2.2 Thermal Cycling-Aware Run-Time Management . . . 100
7.3 Experiment Results and Analysis . . . 101
7.3.1 Block-Level Thermomechanical Stress-Aware Floorplan . . . 102
7.3.2 System-Level Thermomechanical Stress-Aware Floorplan . . 104
7.3.3 System-Level Run-time Thermal Management Scheme . . . 105

    - 7.3.4 Sensitivity Study on TSV Thermal Conductivity and Diameter 106
    - 7.3.5 Impacts of Thermal Through-Silicon Vias 108
  - 7.4 Summary 110

Chapter 8
Conclusion and Future Work **111**

Bibliography **114**

List of Figures

1.1 Conceptual view of TSV-based 3D integration. 3
1.2 Conceptual view of interposer-based 3D integration. 3
1.3 Conceptual view of monolithic 3D integration. 3

2.1 The test cost percentage in 3D integration with different die yields. 14
2.2 The normalized system cost of four cost estimation combinations. . 15
2.3 The overview of the proposed framework for cost estimation and design option analysis. 17
2.4 The variation in partial yield with different defect clustering degrees and two tolerant thresholds. 21
2.5 Both extremely centralized and extremely scattered defect distribution result in high partial yield. (a). Scattered defects; (b). Centralized defects. 21
2.6 Cost variations with different die sizes when the unit count is fixed. 22
2.7 Cost variations with different unit area when the die area is fixed. . 23
2.8 Cost variations with different die sizes/unit counts when the unit area is fixed. 23
2.9 Cost variations with different defect cluster radius. 24
2.10 The impact of reducing wafer/bonding costs on integration costs. . 25

3.1 An example of 3D interposer stacking. (a). Stacking chips are on one side of interposer; (b). Stacking chips are on both sides of interposer. 32
3.2 Cost efficient design exploration flow in 2D, 3D TSV bonding and 3D interposer stacking integrating metal layer reduction technique. 33
3.3 Cost comparison between design implementation on 2D, 2 layer 3D TSV, and 2 layer 3D interposer designs. 36
3.4 Product costs of 2, 3, and 4 layers TSV based 3D designs, 2D and interposer based design costs are shown for comparison. 37

4.1 Overview of core-to-cache/memory 3D stacking. (a). Cores are allocated in all layers with part of cache/memory; (b). All cores are located in the same layers while cache/memory in others. 41
4.2 Timing diagram of the packet latency reduction with additional channel width. (a) Original design with fixed channel width; (b) Designs with additional output channel width. 45
4.3 Virtual-channel based router structure and the corresponding crossbar. (a). Virtual-channel based router architecture; (b). The crossbar design supporting additional vertical links. 46
4.4 Router placements on the core layer for a 16-core system. (a). The interlayer routers are placed in the middle; (b). The interlayer routers are distributed evenly across the core layout. 49
4.5 Packet latency of uniform traffic in a 4×4 mesh network. 52
4.6 Packet latency of interlayer traffic in a 4×4 mesh network. 52
4.7 Packet latency of weighted interlayer traffic in a 4×4 mesh network. 52
4.8 The input buffers read/write ratio analysis of the interlayer router. 55
4.9 The input buffers read/write ratio in different directions of the 2D router. 55
4.10 The dynamic power consumption comparison between baseline and proposed design. 56
4.11 The average packet latency in 8×8 mesh network with the weighted interlayer traffic. 56
4.12 The input buffers read/write ratio of the interlayer router in the middle layer. 57
4.13 The average packet latency with different locations of unavailable spare TSVs and dynamic failure modeling. 58

5.1 The overview of stacked dies with 3D P/G network. 63
5.2 Top view of TSV array with unevenly distributed current. 63
5.3 The EM lifetime analysis framework of TSV arrays consists of current distribution calculation and array lifetime estimation. 65
5.4 The 2D planar P/G grid network model on each tier. 66
5.5 The portion of 3D power supply network with one TSV array (only VDD is shown). 66
5.6 The corresponding resistance path of 3D power supply network. . . 67
5.7 4×4 TSV array lifetime distribution. 72
5.8 EM lifetime distribution with different TSV arrays. The y-axis shows the number of samples/percentile, and x-axis shows the normalized EM lifetime in log scale. 72

5.9 TSV array EM lifetime distribution with different filling materials: Cu, Al, and W. The y-axis shows the percentile of the TTF distribution, and x-axis shows the normalized lifetime in log scale. . 73
5.10 TSV array EM MTTF with different TSV counts in the array under three different current stresses. 75
5.11 TSV array EM MTTF with different diameters under constant current load. 75

6.1 The capacitance crosstalk model in a 3×3 TSV array. 80
6.2 Framework overview of signal transmission with LAT coding scheme. 84
6.3 The overhead caused by redundant TSVs with different ω. The horizontal axis is the bitwidth of dataword and the vertical axis is the overhead percentage. 87
6.4 The overview of signal transmission procedure after overhead optimization. 88
6.5 The benchmark crosstalk characteristic in 3D cases. 93
6.6 The transmission time results of ideal case, ShieldUS, and 6C crosstalk minimization schemes normalized to uncoded case. 94

7.1 Stack level thermal cycling effect in 3D structure. Thermal stresses are pointing from hot blocks(dark color) to cool blocks(light color). Alternating direction of stresses (the arrows) easily cause cracking on thinned substrate. 98
7.2 Run-time thermal cycling-aware thermal management flow for one sampling cycle. 101
7.3 Floorplan of the core-layer before and after optimization. (a) TSV bus is placed horizontally in the floorplan for wire length and area driven floorplan; (b) TSV bus is placed vertically after thermomechanical stress aware floorplan. 104
7.4 Zoomed in TSV thermal stress-aware floorplan on core-layer. (a) TSV bus is placed horizontally between execution units of two cores, resulting in higher TSV temperature; (b) TSV bus is placed vertically to for reduced thermal load. 105
7.5 Zoomed in memory layer run-time thermal management results. The zoomed region causes inverting thermal cycling pattern between two layer. (a) The thermal map before power scaling; (b) thermal map after power scaling on interested region. 106
7.6 Temperatures with different TSV thermal conductivity settings. . . 107
7.7 Temperature results with different TSV diameters. 107
7.8 Thermal maps of benchmark *hp* without thermal vias. 108

7.9 Thermal maps of benchmark *hp* with thermal vias. 109
7.10 Thermal maps of benchmark *hp* when reducing the weight of wire length without thermal vias. 109
7.11 Thermal maps of benchmark *hp* when reducing the weight of wire length with thermal vias. 110

List of Tables

2.1 Multi-core processor design system cost comparison 25

3.1 Parameters used in the experiments 36
3.2 Results of optimal designs after cost efficient optimization process for 2D, TSV and interposer based 3D. 37
3.3 Cost Savings of 2D, TSV and interposer based 3D designs for optimal area utilization with default metal layers. 38
3.4 Cost Savings of 2D, 3D TSV bonding, and 3D interposer stacking designs with metal layer reduction technique. 39

4.1 The synthesized area and power of 2D/3D routers, crossbar, and AL arbitrator. 51
4.2 Average packet latency comparison of different layers in 3D systems. 57

5.1 EM MTTF with fixed area budget. 76

6.1 Coding overhead comparison results of with and without optimization. 89
6.2 The power consumption related parameters comparison between uncoded, ShieldUS, 3D 6C CAC, and 3-LAT schemes. 92

7.1 Thermal parameters that are used in the experiments; sensitivity study is performed with various TSV lateral thermal conductivity and TSV diameter. 102
7.2 Design time thermomechanical stress-aware floorplan results with average and peak on-chip temperature and thermal stresses reduction 103
7.3 The normalized area/wire length and execution time comparison . . 103

Acknowledgments

First of all, I would like to express my sincere gratitude to my advisor Professor Yuan Xie. He introduced me to the world of computer engineering and encouraged me to pursuit my Ph.D. degree when I was confused about my future. His unconditional help and patient guidance throughout all these years make my Ph.D. study more smooth and impel me to conquer the problems in hard times. His positive attitude and passion towards research and working inspire me and become one of the important factors to stimulate my growth in both research and personality.

I would like to thank all my committee members as well, Professor Mary Jane Irwin, Professor Vijaykrishnan Narayannan, and Professor Donghai Wang, for their precious advices and feedback on my research work.

I would also like to thank all my colleagues and friends from MDL group for their kind suggestions and supports on both my life and research: Xiaoxia Wu, Xiangyu Dong, Jin Ouyang, Cong Xu, Jishen Zhao, Matt Poremba, Jia Zhan, Ping Chi, Ivan Stalev, Yang Zheng, Shuangchen Li, and Ziyang Qi. Especially, I owe my thanks to my roommates, Hsiang-Yun Cheng, Jue Wang, Lin Chen, and Zhuo Chen for providing me a such warm and cozy living and study environment. Also, I want to thank my co-authors, Yibo Chen, Dimin Niu, Tao Zhang, and Jing Xie, as they gave me tons of valuable suggestions on my research work. Besides, I want to thank all my Ph.D friends from other departments: Chu Huang, Yiqiang Han, Yan Cao, and Zhe Liu, for their companion. During my intern at Huawei Shannon Lab, I learned a lot and got plenty helps from my collaborators, including but not limited to Rui He, Xing Hu, Shihai Xiao, and Wei Yang.

Finally, I owe my deepest gratitude to my family. I thank them for their understanding and supportive love for my every wilful decision. Especially, I want to express my heartfelt thanks to my husband, Wei Zhou, for backing me up across the ocean for these many years without a single complaint.

Chapter 1

Introduction

In this chapter, the background of three-dimensional integrated circuits (3D ICs) is first introduced. Then the state-of-art design methodologies of 3D ICs are discussed followed by the challenges that hinder the massive volume production in the industry. The organization and contribution of the remaining chapters are then given.

1.1 3D Integration Technologies

With technology scaling, the continuously decreasing transistor size implicates higher integration density [1]. Inevitably, this integration density translates into ever-growing interconnect length. Moreover, the scaled interconnect parasitics (smaller wire cross sections and smaller wire pitch) makes the interconnect emerge as a dominant source of circuit delay and power consumption [2, 3]. The solution of the interconnect crisis is of paramount importance for deep-submicron designs. The 3D integration technology is one of the favorable solutions as it provides the benefits of short vertical interconnects, smaller footprint, and capability of heterogeneous stacking [4–6].

In general, there are three types of 3D designs in terms of the interconnect technologies: system-in-package designs, fine-grain integration, and contactless stacking. The first kind of integration can utilize the traditional 2D designs and connect the signals and I/Os through low density, relatively long vertical connections, such as wires or peripheral solder bumps. The second requires additional process steps to build the vertical connections insider the device chip, thus it can provide higher density connections with lower latency. The third technology leverages the inductive or capacitive coupling for signal transmissions. The fine-grain 3D integration is attracting a growing number of popularity as it enables high density of vertical connections compared to the SiP and is free of distance restriction as compared to the contactless integration. As a consequence, this dissertation focuses on the design methodologies of this kind of integration.

Specifically, the fine-grain integration can be conceptualized either as a parallel or a sequential process. In the parallel process, each individual dies are fabricated in parallel and through-silicon vias (TSVs) are built for signal connection between tiers. The wafer is then thinned to expose the TSVs and bonded to form a stack, as shown in Figure 1.1. Recently, integration circuits with TSV-based interposer (also known as 2.5D ICs, Figure 1.2) are emerging as alternatives of 3D ICs due to the simpler fabrication process and ease of cooperation between companies. In the parallel process, since the bonding is performed after individual fabrications, there are different bonding orientations and stacking methods. In terms of bonding orientation, it can be classified into face-to-back (F2B), back-to-back (B2B), and face-to-face (F2F) bonding, in which, the face represents the side with active devices and the back is the bulk silicon. For F2B and B2B stackings, TSVs are required for signal and I/O connections, while in the F2F bonding, the micro-bump is utilized to connect the topmost metal layers of two dies for signal communications, and TSVs are used for I/O connections. The stacking can be realized in three strategies: die-to-wafer (D2W), die-to-die (D2D), and wafer-to-wafer (W2W), which is distinguished by whether the die is sliced before it is stacked. In general, W2W stacking can shorten the time-to-market, however, it has the lowest compound yield as any bad dies in the stack will pollute the final stack. D2W and D2D stacking can prevent the bad dies from polluting good stacks, nevertheless, the requirement for testing of each individual dies increases the system cost and prolongs the time-to-market.

In contrast, the sequential process, also named as monolithic 3D ICs (Figure 1.3), requires the device layers to fabricate sequentially and the Monolithic Inter-die Vias (MIVs) are built with traditional back-end-of-line (BEOL) technology. The small size of MIVs supports higher density integration at finer granularity

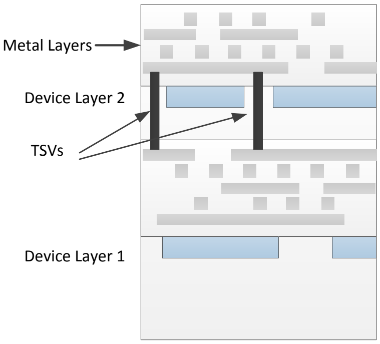
Metal Layers
Device Layer 2
TSVs
Device Layer 1

Figure 1.1. Conceptual view of TSV-based 3D integration.

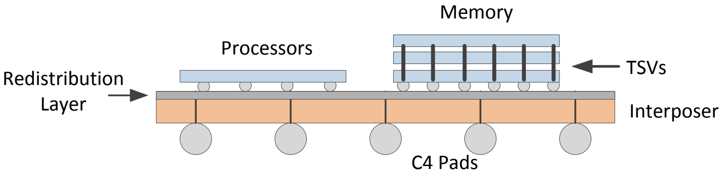
Memory
Processors
TSVs
Redistribution Layer
Interposer
C4 Pads

Figure 1.2. Conceptual view of interposer-based 3D integration.

3D partitioning, even at the granularity of transistor. The sequential process poses numerous challenges in 3D monolithic design since the high temperature during fabrication compromises the device mobility. As a consequence, before the technology becomes viable and reliable for complete 3D monolithic, the parallel 3D integration is the precursor.

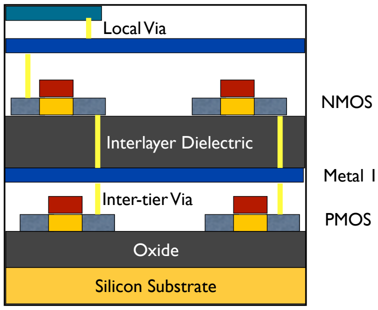
Local Via
NMOS
Interlayer Dielectric
Metal I
Inter-tier Via
PMOS
Oxide
Silicon Substrate

Figure 1.3. Conceptual view of monolithic 3D integration.

1.2 Existing Physical Design Methodologies of 3D ICs

Due to the additional third dimension, the circuit design cannot be directly migrated from 2D ICs. Innovated design methodologies are necessary to fully unleash the benefits of 3D ICs. In this section, the existing physical design methodologies targeting 3D ICs are introduced.

1.2.1 3D Partitioning and Floorplanning

Partitioning. The partitioning step is unique in 3D designs and it has great influence on the following floorplanning quality. In this design stage, we need to identify the 3D integration technology, the partitioning granularity, and the design objective. Generally speaking, the partitioning can be done on fine granularity (gate level), or on coarse granularity (block/core level). With smaller vertical connections (e.g. 3D monolithic), finer granularity partitioning can be realized. Different partitioning mechanisms result in various design benefits [7–13]. For example, Loh et at. [14] proposed three cache partitioning schemes and compared the differences in performance improvement and latency/energy reduction.

Floorplanning. To enhance the design flexibility, the layer position is usually determined during floorplanning, which defines the placement in the block granularity. In this stage, the blocks can be moved within one layer or between layers to obtain the optimal result in terms of area or wirelength.

The generalized problem formulation of floorplanning can be described as below. Given a set of modules/blocks with the geometry information of each component, the allowable aspect ratio during rotation, and the total number of layers; based on the design objective, determine the coordinate of each block. According to the design objective, additional information may be needed. For example, when the temperature is more important, the power density of each block should be considered.

Various floorplan representations in 2D ICs can be used in 3D designs, such as sequence pair and B*-tree. Meanwhile, new floorplan topologies are denoted with novel representations [15, 16]. After the floorplan representation and the design objective are determined, the floorplan algorithm manipulates the block position until the optimal result is found. Numbers of floorplanning strategies targeting on different design objectives have been proposed [17–22]. Jung et al. [23] studies the impact of block folding and bonding styles on top of traditional 3D floorplanning to improve the power benefits of 3D ICs. Their design offered up to 20.3% power reduction over the 2D counterpart under the same performance. Tsai et al. [18] emphasized the TSV planning on accurate wirelength estimation during the floorplanning. Falkenstern et al. [24] used B*-Tree for representation and consider the power/ground network synthesis during floorplanning to reduce the potential IR drop. Khan et al. [15] proposed a new topological structure for floorplanning in 3D ICs to minimize the total volume of 3D die.

1.2.2 3D Placement and Routing

Placement Based on the block position decided by floorplanning, the cell/gate position is determined during placement. Different from 2D designs, the 3D placement not only needs to consider the cell spreading on the xy plane, but also in the vertical direction. If temperature is the primary concern, then cell placements on the upper layers should be sparser than the bottom layer which is near heat sink. Temperature and wirelength are two major design objectives during placement.

Various of placement methodologies are presented by previous studies [25–29]. Cong et al. [30] proposed to use transformation for 3D placement with the awareness of temperature. In their proposed framework, a 2D placement is first generated with specific design objective. Then local stacking transformation and a folding-based transformation are used to change the 2D placement into 3D placement. A 3D level refinement is performed to guarantee the design quality. Athikulwongse et al. [31] proposed two thermal-aware global placement algorithms employing the force-directed methodology to exploit the die-to-die thermal coupling in 3D ICs. The first algorithm generates forces for TSV spreading and alignment for better heat dissipation path. The second algorithm builds forces based on thermal conductivity in cells and on power density for TSVs. Their second methods can achieve the best temperature results among state-of-the-art placers. An analytical placer is proposed by Cong et al. [32]. They took the thermal effect of TSVs into consideration and co-placed cells and TSVs. Experimental results show that their method can reduce the peak temperature by 34% on average.

Routing. In 2D routing, optimization targets usually include wirelength, the interconnect delay, and the routing congestion. When moving to the 3D design, more issues should be considered. First, the interconnect delay is closely related to its temperature, therefore, wires on the critical path should avoid the hot regions, moreover, the interconnect placement should consider help alleviate the on-chip temperature [33, 34]. In addition to the temperature, other factors also influence the interconnect, such as thermal stresses, which should be addressed in the routing stage for better and robust performance [35]. The second problem rises from the limited number of TSVs. Due to the large size of vertical interconnects, TSVs are scare resources and should be optimally allocated. Third, the congestion management and blockage avoidance become more complicate, especially when considering the thermal via insertion [36, 37]. The routing on each tier is similar to the traditional 2D routing, only extended to the third dimension with limited vertical routing capacities.

The 3D routing begins from building a minimum spanning tree or steiner tree similar to 2D routing. Then a multi-objective routing algorithm is applied to guide the routing. The associated signal TSV and thermal TSV assignment are commonly accomplished with two approaches: a heuristic multilevel approach and a linear programming (LP) based approach [38–40]. Two stages of TSV positioning are performed to first locate TSVs according to Steiner tree considering the wirelength. Then the stacked-TSV relocation stage assign stacked TSVs near hotspots for temperature reduction [38]. The stacked-TSV assignment exhibits 17% temperature reduction with 4% wirelength overhead and 3% performance degradation. Pathak et al. [39] employed the similar idea as moving TSVs close to hotspots to reduce the cost of dummy TSVs insertion after the initial tree construction. They proposed a relaxed ILP-based formulation to optimize all nets simultaneously. A 9% maximum-temperature reduction is gained with their TSV relocation. At last, the level-by-level routing refinement is performed to alleviate the congestion.

1.3 Existing System Level Design Methodologies

In addition to the physical level design methodologies, plenty of research efforts are devoted into the system level or the high level design methodologies of 3D ICs. Different from physical level designs, the system level designs focus more on the holistic optimization and whole system design tradeoffs in accordance with design requirements. It sometimes involves the architecture-level or system-level design space exploration to find the optimal design point [41–43].

The first possible direction of system level design space exploration is incorporated with high-level synthesis techniques [44–47]. In the high-level synthesis, the system behavior is described in natural language or high level programming language, and various related components are provided as candidates to assemble an optimized system. These components can be either hardware solutions or software solutions, and usually, the same functionality has different implementations. Different from previous 2D designs, the 3D integration synthesis should take the benefits of vertical connections into consideration, and might incorporate with detailed physical level design techniques as demonstrated by Chen et al. [45]. Another methods in system level design is to chain the available EDA tools to form a holistic design tool chain [48].

In addition to the above-mentioned two methodologies, the architecture level 3D integration is examined by various studies focusing on the 3D logic-to-logic stacking in the multicore processor designs or even FPGA designs [49–51], or the more prevalent memory-to-logic stacking to solve the memory bandwidth issues [52–55]. By stacking computing or communication fabrics on top of others brings close proximity, results in advanced performance, lower latency, and energy efficiency. Even though 3D designs can improve system performance without substantial architecture modifications, Loh pointed out that by implementing only a few simple changes to the commodity DRAM organization, a 1.75x speedup can be achieved against baseline 3D DRAM [52].

Besides the homogeneous architecture designs, the heterogeneous system design space is explored to unleash the benefits of 3D ICs [56, 57]. The heterogeneous integration can build from four aspects: different on-chip network components, such as electrical and optical connections [58–60]; different memory technologies, such as traditional SRAM/DRAM with non-volatile memory [61–63], different computing logics, such as general purpose CPU, GPUs, and accelerators [64–66]; different signal handling systems, such as analog circuits and digital circuits [67, 68].

1.4 Opportunities and Challenges in 3D ICs

The advantages of 3D ICs attract numerous architecture researchers' attention to solve the prominent Memory and Power Wall problems. Lots of literatures are presented in recent high quality conferences, such as using stacked DRAM as cache [69] or as on-chip memory for high bandwidth computing [53]; or stacking heterogeneous computing/storage fabrics for energy-efficient operations [70].

In addition to academia, the commercial adoptions of 3D memory and 3D FPGA have demonstrated the promising future of 3D integrations. For example, the High Bandwidth Memory standard as proposed by JEDEC [71] can provide as high as $256GB/s$ bandwidth, targeting on graphic applications. The Hybrid Memory Cube system is proposed to support multiple stacked DRAM through SerDes links and the maximum bandwidth can reach $320GB/s$ [72]. Moreover, the 2.5D design has been applied to Xilinx Virtex-7 FPGA to enable high-bandwidth connections between multiple dies and provide a 100× improvement in inter-die bandwidth per watt [73].

Even these examples from both academia and industry show promising future of 3D ICs, there are still some stoppers before massive production of more sophisticated 3D circuits, such as the thermal management, reliability problem, testing mechanisms, and cost issues.

The thermal problem in 3D ICs exacerbated due to two reasons: the reduced proximity between active devices and the increased distance from the bottom devices to the topmost heat sink. Numerous solutions are proposed to boost the thermal dissipation capability, such as incorporating with more powerful cooling solutions [74–76] and inserting dummy TSVs for building effective vertical dissipation path [38, 77]. The reliability problem stems from either manufacturing limitations or design issues. The manufacturing constraints introduce new circuit failures and fault models, such as TSV open/short defects, thermal mechanical stresses, and device mobility variations [78–82]. Moreover, design level considerations should include signal integrity, power/ground network with less IR droop, and balanced clock network designs [83–88]. The reasons that testings become critical and challenge in 3D ICs can be classified into three categories: new defect modes as in the reliability problem, incomplete functionality on each individual layer, and lack of testing methodologies (pre-bond, intermediate, and post-bond tests) [89–93]. Cost is the primary driving force for industrial adoption of 3D ICs. However, due to the extra wafer thinning and bonding processes, the fabrication cost of 3D ICs is not always lower than that of 2D counterparts. Moreover, in consideration of the re-design and testing efforts, the ever increasing non-recurring cost further worsens the final selling price [94–96].

Currently, most researchers focus on the physical and system level design methodologies to reduce the wirelength and chip area, addressing the power and memory wall problems. However, the prominent challenges in 3D ICs are not properly solved, especially the thermal and reliability problems. Therefore, in this dissertation, reliability and cost are the two primary concerns in the proposed 3D design methodologies and optimizations.

1.5 Contribution and Organization

In this dissertation, the design methodologies and optimizations of 3D ICs are proposed to consider the integration cost and reliability issues. Different from previous studies that ignore the impact of cost in practical designs, the proposed cost-aware design methodologies are easy for industrial adoption. Moreover, the proposed reliability-aware design approaches perform reliability analyses and give circuit/architecture level implications from various reliability aspects (electromigration, thermo-mechanical stresses, etc.)

The following chapters are organized as following. Based on the 3D cost model, Chapter 2 proposes a cost evaluation framework with the assumption of test elimination given the assumption that the fabrication yield is sufficient high. Chapter 3 reviews the routing process and the cost of building metal layers, then describes the aggressive metal layer reduction technique to sacrifice chip area for cost saving from less metal layers. The cost saving technique to reuse the redundant TSVs can be applied on the architecture level for NoC designs, as introduced in Chapter 4.

Three studies are selected towards the reliability problems. Chapter 5 reconsiders the electromigration (EM) problem given that TSV array is usually used for signal transmission instead of a single TSV. The EM lifetime in an array should be different from the analyses of an isolated TSV. Chapter 7 takes the thermo-mechanical stresses into consideration, and proposes a two-stage thermal stress management scheme in both design-time and run-time. The large size and close proximity of TSVs pose challenges in signal integrity due to the capacitance coupling. Chapter 6 analyzes the crosstalk phenomenon in the TSV array and a coding-based crosstalk minimization technique is then proposed. At last, Chapter 8 provides the conclusion and the future work.

Chapter 2

Cost Analysis Framework with Test Elimination

As illustrated in Chapter 1, several shortcomings hinder the commercial adoption of 3D designs. Among these challenges, the 3D integration cost (both recurring and non-recurring cost) is one of the dominant factors [94–96] since the profit margin is the driving force for industrial migration to 3D designs. In this work, we propose to reduce the final system cost by exploring the possibility of test elimination given that the die yield would be sufficient high as fabrication process improved[1].

In the final system cost breakdown, test cost stands out as one of the major contributors [94,97,98]. In addition, as the CMOS manufacturing process becomes mature, the fabrication cost decreases, which results in a higher percentage of the test cost. Moreover, in 3D designs, testings pose numerous challenges due to the insufficient understanding of 3D testing issues and the lack of efficient testing solutions. There are two major reasons lead to the testing differences between 3D designs and 2D counterparts. The first reason is that any given layer may contain uncompleted modules (e.g., networks for power delivery and clock distribution) due to the function partitioning in 3D ICs. The thinned wafer and the resulted mechanical vulnerability are the second reason. Wafers need to be thinned to expose vertical vias (i.e. TSVs). However, the wafer level probing force, which is about 60-120kg per wafer [99], may damage the thinned wafers.

---

[1] This work is published as "A cost benefit analysis: the impact of defect clustering on the necessity of pre-bond tests" on 3DIC 2015.

Pre-bond and intermediate tests are unique in D2W and D2D stackings. Previous studies have shown that cost saving can be obtained through the pre-bond or intermediate testing because it eliminates further process steps on failed stackings [98, 100, 101]. However, these observations are based on the wafer yield which may be underestimated through the Poisson model or the Generalized Negative Binomial (GNB) model [102]. With the presence of defect clustering [103, 104], the die yield should be determined by both the defect count and the defect distribution. If most defects are clustered in a small region, die yield could be significantly higher than the case with random distribution, even though the defect rates are the same. Therefore, under this circumstance, potentially we can eliminate the pre-bond and intermediate testings without elevating the integration cost.

To further reduce the integration cost, we also investigate how the partially functional dies help the system cost. In order to provide advanced and robust performance, multiple disjoint units with the same functionality might be placed on a single die (e.g., homogeneous multicore designs). *Partially functional die* represents the die that contains some workable units even one or several units are malfunctioned. Rather than simply discarding them, we can sort out the slightly defective dies for lower price according to the design expectation, that is, the expectation of the number of functional units per die. Thus, the traditional concept of bad die is changed to partially functional die for homogeneous multi-unit designs. Therefore, *partial yield* is defined as the ratio of defective dies that can fulfill the design expectation accordingly.

To help manufactures and designers make testing decisions at the early design stage, we build a cost benefit analysis framework. In addition to the testing elimination decision, the framework also provides the estimated system cost using corresponding analytical models. To support the proposed framework, we further provide a hierarchical yield model to estimate both die and partial yields. We reveal the analytical model for partial yield by investigating the relation between partial yield and the density of defect clustering. With this framework, we discuss how different cost parameters influence the system cost. From the analyses, we find that wafer cost is the most significant factor on test elimination and cost saving.

2.1 Preliminaries and Motivation

In this section, the preliminaries of die yield calculation and the motivation example are demonstrated. Previous work on 3D cost analyses and yield enhancement are briefly introduced.

Preliminaries on yield modeling. GNB yield model, as shown in Equation 2.1, is prevalently used in previous studies for the depiction of defect clustering, when the defect density (D_0) and die area (A) are given. By adjusting the parameter α, different clustering degrees can be expressed. The smaller value of α, the higher clustering degree.

$$Y(GNB) = (1 + \frac{D_0 A}{\alpha})^{-\alpha} \tag{2.1}$$

Even the GNB yield model considers the defect clustering to some extent, it cannot accurately capture the die yield under extreme clustered defects. Moreover, the value of α is hard to determined even given the defect distribution on a wafer.

Motivation. As manifested from previous studies [96–98], the test cost is one of the major contributors of the 3D system cost. We assume a 4-layer D2W stacking is used, and vary the test cost (increasing the test cost up to 4 times) and die yield to illustrate the ratio of test cost. The cost parameters (value of test cost, wafer cost, etc.) are consistent with the prior study [98]. The cost estimation is based on previous 3D cost model [96] and the results are depicted in Figure 2.1. When the die yield increases, which is likely to happen due to the continuous evolvement of fabrication process, the test cost increases almost linearly. The growing step becomes larger with higher testing costs. For example, when the testing cost is four times of the original value, the percentage increases more than 3%, while in the original case, the percentage increases less than 1%. Moreover, according to the observations in [97], the testing cost percentage increases with more stacking tiers. In this example, the testing circuit overhead is not included, resulting in an underestimated test cost. If the pre-bond/intermediate test is performed for every die and every intermediate stacking in D2W integrations, the circuit overhead for testing makes the test cost unaffordable.

The compound yield of the whole stacking would influence the selection of test flows. Therefore, we want to study when the die yield is high, what would happen

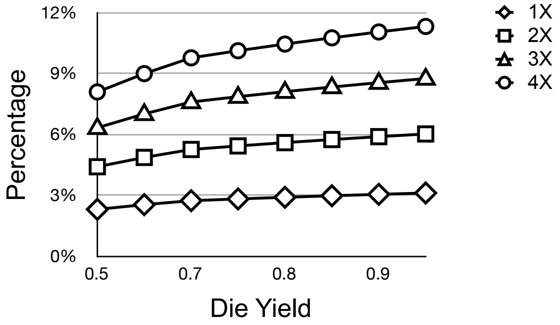

Figure 2.1. The test cost percentage in 3D integration with different die yields.

to the system cost if we eliminate the testing. Assume a wafer has a defect rate of 0.5 defects/cm^2 and a diameter of $300mm$, and the die area is $50mm^2$. The calculated yield with GNB model is 81.65%. The compound yields are as low as 66.67%, 54.43%, and 44.45% when the stacking contains 2 tiers, 3 tiers, and 4 tiers, respectively. With our die yield simulation, which generates defect maps with given clustering degree, the statistical die yield can be as high as 97.6%, marked as the simulated yield.

Based on the assumption above, we conduct a study to find out the cost influence when eliminating pre-bond/intermediate testings. The integration system costs of four cases are shown: GNB model with tests (basic case), GNB model without tests, simulated yield with tests, and simulated yield without tests. Figure 2.2 shows the system costs of 3 and 4 layers stackings which are normalized to the cost of 2-layer basic case. We observe that the cost saving from only manufacturing yield improvement is not so obvious when comparing cases 1 and 3 which are with tests. However, when comparing cases 2 and 4, dramatic cost saving is achieved with higher die yield. Moreover, when the die yield is high enough (case 3 and 4), the estimated cost difference between testing and testing elimination scenarios becomes negligible, meaning that the cost saving from testings is marginal. These results reveal the opportunity of eliminating pre-bond/intermediate tests when the yield is high. The test elimination poses negligible cost overhead yet can shorten the time to market.

Prior Work. The 3D test cost and system cost have been modeled and analyzed by previous studies [96–98]. Numerous cost-efficient designs and optimization flows are explored [94, 95, 105]. Researchers find that the selection of testing flows

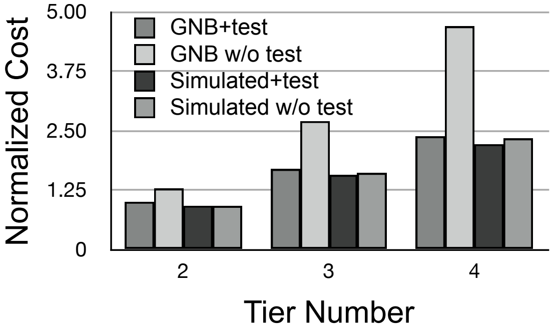

Figure 2.2. The normalized system cost of four cost estimation combinations.

has significant impact on the final integration cost [94, 98]. The 3D testing can be categorized into four steps: pre-bond tests to filter good dies; intermediate tests to ensure the functionality after die bonding; pre-package tests for the whole 3D integration before package; and post-package tests for the product integrity. The pre-bond and intermediate tests are unique in the die-level stackings, while pre-package and post-package tests are essential for all 2D and 3D designs. In this work, when we consider eliminating the test, we refer to pre-bond and intermediate tests. The later two tests cannot be eliminated since the final system functionality should be guaranteed before marketing.

In addition to testing, wafer/die matching and redundancies sharing between tiers are two major yield enhancement methods [90, 106–109]. Usually, three metrics can be used during the matching process: maximized the matched good dies, maximized the matched bad dies, or minimized the unmatched bad dies. In the wafer matching method, the wafer are carefully selected from wafer repositories to increase the yield. The method of die matchings considers the redundancy resources to guarantee redundancies are enough to share between dies. As such, the final compound yield is increased. The limitations of these studies are that they diminish the product throughput and current algorithms only handle the matching between two layers. However, these mechanisms are orthogonal to our work and can be combined for further cost saving.

2.2 Cost Benefit Analysis Framework

As demonstrated in Figure 2.2, when the die yield is high enough, the pre-bond or intermediate tests can be eliminated for cost saving. Furthermore, the partially functional die can be applied to gain more profit for amortizing the cost. However, various design tradeoffs may affect the cost saving. When the bonding cost or wafer cost decreases, can the design still get benefits from the test elimination? What is the die yield turning point for pre-bond test elimination? How do the size and arrangement of homogeneous units affect the system cost? To answer these questions at an early design stage, we propose and implement a cost benefit analysis framework to select the cost-efficient design.

Our framework can be depicted in Figure 2.3. The inputs contain three parts: the defect mode, cost related parameters, and the design target.

Defect Mode. The defect mode is used to generate defect maps for the hierarchical yield model. The information of defect mode includes wafer defect rate, the defect clustering degree, and the yields during bonding and packaging. In our framework, we use the cluster diameter (generated by clustering algorithms from manufactures) to describe the clustering degree. At the early design stage, the detailed cluster size for each wafer cannot be obtained. Therefore, an estimated average value is enough for the cost evaluation. In addition to fabrication processes, defects can occur in the bonding and packaging steps in 3D designs. Wafers need to be thinned to around $100\mu m$ to expose TSVs, which makes wafers vulnerable to mechanical stresses. Moreover, due to the manufacturing limitation, the TSV yield is beyond expectation. Plenty of work has been done to improve the TSV yield with redundancies [100, 110], which significantly improves the bonding yield.

Cost Parameters. For 3D integrations, the system cost can be classified into four categories: wafer cost, bonding cost, testing cost, and package cost [96, 97]. The wafer cost is the major contributor to the system cost. The bonding cost is unique in 3D stacking, which contains the wafer thinning cost, TSV building cost, and bonding cost. The test cost includes all recurring and nonrecurring expenses in pre-bond and intermediate tests. The pre- and post-packaging costs are captured in the package cost category.

Design Target. The defect tolerance and unit design options should be given

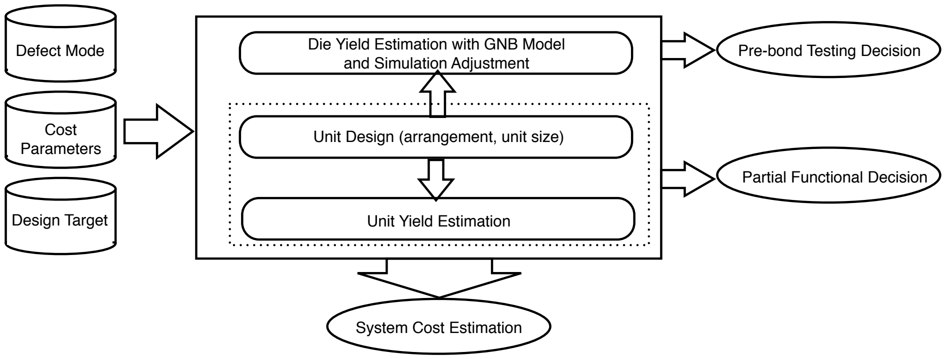
Defect Mode
Cost Parameters
Design Target
Die Yield Estimation with GNB Model and Simulation Adjustment
Unit Design (arrangement, unit size)
Unit Yield Estimation
Pre-bond Testing Decision
Partial Functional Decision
System Cost Estimation

Figure 2.3. The overview of the proposed framework for cost estimation and design option analysis.

in the design target. As illustrated in Section 2.3, the tolerant threshold can significantly influence the partial yield. Instead of giving the determined design, we require design options in our framework due to the impact of unit size and arrangement on the partial yield and die yield. Take the multi-core design as an example, if a total of 8-issue width design is needed, we can use four 2-issue width cores or two 4-issue width cores with different unit and die areas. Even when the die area is fixed, different unit arrangements influence the partial yield. For example, if the design of four 2-issue width cores is selected, placing the cores in 2×2 or 1×4 arrangement results in totally different unit yields. Therefore, holistic design consideration should be made considering both the die yield and partial yield. However, the enormous possible design combinations make the design space exploration infeasible. Fortunately, due to the thermal and power constraints for each design, the design options are limited, and they should be provided as an input in the framework.

The main body of the framework takes these three inputs and generates the cost-efficient testing and design decisions and the corresponding cost estimation. Multiple testing schemes are possible for any 3D stacking [94]. For example, we can choose to test circuit only, interconnect only, or circuit and interconnect during the intermediate test. In this paper, we only consider one testing scheme, which is the pre-bond test for every die and the intermediate test for both device and interconnect whenever a bonding process is established. Although only one test scheme is assumed in this work, other schemes can be easily applied in the framework. The cost saving of partially functional die is determined by the partial yield which is influenced by the unit size and arrangement. Therefore, by varying the unit design, the die yield and partial yield are updated accordingly until a cost-efficient design is found. Then the die area is fixed, the die yield can be calculated from the combination of GNB model and simulation results or the defect model provided by the manufacture. The decision that whether performing the pre-bond/intermediate test or not can be made given the die yield and cost parameters.

After design decisions are determined, the cost evaluation is performed. According to [95, 96], the cost of a N-layer D2W stacking can be captured with Equation 2.2 combining with the pre- and post-packaging test. In the equation, C_{die_i} represents the die cost on tier i, which can be calculated using C_{wafer}/N_{die_i}. C_t, C_b, and C_p are the test cost, bonding cost, and package cost, respectively. Y_p is the package yield determined after pre- and post-package testings.

$$C_{D2W} = \left(\frac{\sum_{i=1}^{N}(C_{die_i} + C_t)}{Y_{die_i}} + (N-1)C_b + C_p \right) / Y_p \tag{2.2}$$

If pre-bond and intermediate tests are eliminated, the expression of system cost is shown in Equation 2.3. Different from Equation 2.2, the bonding cost should be considered for every die (including defective dies). The corresponding compound yield, which is the multiplication of die yield in each tier, is applied. In this sense, the system cost for D2W and W2W stacking are similar. Note that in W2W designs, the die size in each layer should be the same.

$$C_{notest} = \left(\frac{\sum_{i=1}^{N} C_{die_i} + (N-1)C_b}{\prod_{i=1}^{N} Y_{die_i}} + C_p \right) / Y_p \tag{2.3}$$

When the die yield is not high enough or the bonding cost is relatively large, the test elimination causes significant cost loss. Therefore, we deduct the system cost equation combining the test elimination and partially functional die, which is shown in Equation 2.4.

$$C_{partial} = \left(\frac{\sum_{i=1}^{N} C_{die_i} + (N-1)C_b}{\prod_{i=1}^{N}(Y_{die_i} + Y_{p,th}(1 - Y_{die_i}))} + C_p \right) / Y_p \tag{2.4}$$

The partially functional die can also be applied even when testings are enabled.

In this case, we only need to add the partial yield to the Equation 2.2. Combining Equation 2.2 and 2.3, the turning point of die yield for test elimination is estimated as follows, assuming the die yield and die cost for each tier are the same:

$$C_t Y_{die}^{N-1} \geq N C_{die}(1 - Y_{die}^{N-1}) + (1 - Y_{die}^{N})(N-1)C_b \tag{2.5}$$

Induced from the equation, the cost parameters (bonding cost and wafer cost) play an critical role in the turning point. Therefore, when the test cost is relatively high compared to the bonding cost and die cost, it would be cost-efficient to eliminate the pre-bond/intermediate test.

2.3 Hierarchical Yield Model

In this section, we describe our hierarchical yield model that comprises two levels. The first level combines the GNB model and simulation results for die yield estimation with the presence of defect clustering. The second level model estimates the fine-granularity partial yield under the influence of defect clustering.

Defect clustering representations have been studied by several work [103, 104, 111]. These models are suitable for the post-fabrication analysis, but not for cost estimation when the detailed defect information is unknown.

We consider different wafer defect situations given the wafer defect rate and defect clustering degree. The result shows that GNB model underestimates the die yield because of the inaccuracy in capturing the defect clustering. Therefore, apart from GNB model, we built a simulation framework to adjust the die yield estimation according to the defect clustering degree. The simulation framework generates massive wafer samples and the corresponding defect maps to mimic the defect distribution statistically. Combining the simulated yield ($Y(sim)$) and GNB yield, the die yield estimation is given by $Y_{die} = \omega_1 Y(GNB) + \omega_2 Y(sim)$, where ω is the weighting parameter to represent the model confidence.

Nevertheless, the first level model is not adequate to determine the system cost when taking partially functional dies into account. Based on the definition, the partial yield is the ratio of the workable defective dies to the total defective dies. The number of reusable defective die depends on the design expectation, the defect clustering degree, and the unit area. Intuitively, the larger the unit area is, the higher partial yield because fewer units may be affected by defects. Meanwhile, if the defect clustering degree is low, the distribution is more likely to be random, thus more units will be defective. Based on the above reasoning, we assume the partial yield $Y_{p,th}$ is proportional to the unit area A_u and inversely proportional to the clustering degree C_r. The variable th represents the tolerant threshold (design expectation) that is determined by designers to represent how many malfunctioned units can be tolerated. By increasing the threshold, more defective dies can be used, however, the selling price for the severe defective dies may be low, which may degrade final profits.

With our simulation framework, the value of $Y_{p,th}$ is simulated with different defect clustering degrees and tolerant thresholds as shown in Figure 2.4. In the simulation, we assume the aspect ratio of the unit (the ratio of the unit width to the unit height) is fixed. The defect clustering degree is represented as the cluster diameter in the unit of mm. We set three levels of unit size: large, medium, and small, which means there are 4, 6, and 9 units per die (die size is $50mm^2$), respectively. Two tolerant thresholds (1 and 2) are used.

From Figure 2.4, we can see the partial yield decreases significantly as the defect clustering degree increases at the beginning, then it increases gradually until stable. The rationale behind this is that when the defect clustering degree increases first, it has a higher probability that more units inside the die (note that the die area is fixed) are affected by the defects. However, when the clustering degree continuously increases, the defects are more disperse until they are nearly randomly distributed in the whole wafer. Therefore, given one die is defective, the probability that only a certain number of units (represented by th) are affected is high. In this case, the die yield is low whereas the partial yield is high. This phenomenon can be explained through a simple example shown in Figure 2.5. We only demonstrate four dies per wafer and four units per die for simplicity. Even though these two wafers contain the same number of defects, defects in Figure 2.5(a) are more scattered, resulting in a lower die yield. However, inside each die, mostly only one unit is affected and the partial yield can as high as 75%. For the concentrated defects as in Figure 2.5(b), the overall die yield and the partial yield are both high. It can conclude that the partial yield is influenced by the area ratio between units and

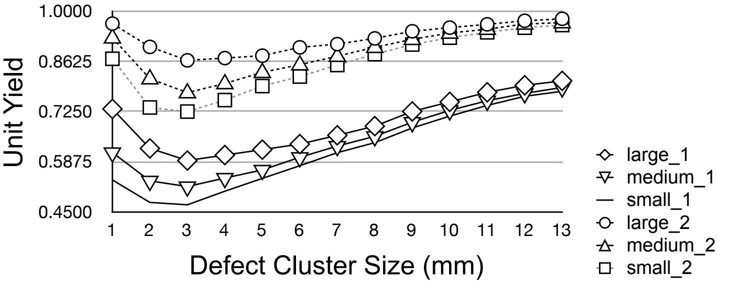

Figure 2.4. The variation in partial yield with different defect clustering degrees and two tolerant thresholds.

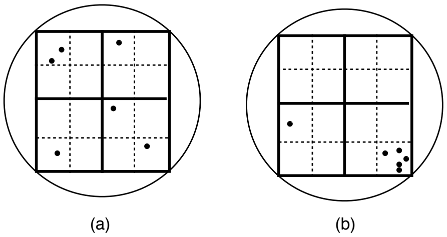

Figure 2.5. Both extremely centralized and extremely scattered defect distribution result in high partial yield. (a). Scattered defects; (b). Centralized defects.

defect clusters, we find the best fitting polynomial curve to analytically represent $Y_{p,th}$ as the following:

$$Y_{p,th} = -k_1R^3 + k_2R^2 - k_3R + k_4 \tag{2.6}$$

where R represents the ratio of A_u/C_r^2. k_1 to k_4 denote the empirical coefficients. The values of k_1 to k_4 are 0.0006, 0.0160, 0.08, and $th/2$, respectively, according to the simulation results in Figure 2.4. This equation can capture the partial yield with the maximum error of 7% with various configurations. The equation is of higher accuracy when the size of the defect cluster is small.

2.4 Experiment and Cost Analysis

In this section, impacts of different design inputs on the system cost are evaluated. These design inputs include the variations of design parameters (the die area, unit area, and unit arrangement), cost parameters[2], and the clustering degree. For

---

[2] Arbitrary unit (a. u.) is used for cost values due to the non-disclosure agreement with our industry partners.

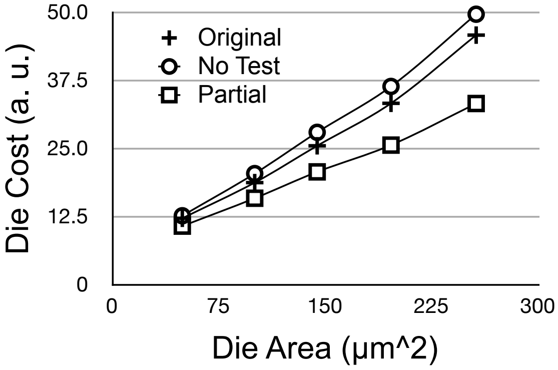

Figure 2.6. Cost variations with different die sizes when the unit count is fixed.

each set of experiments, costs in three scenarios are used for comparison: the basic case, which uses the GNB yield model with the pre-bond and intermediate test; the non-testing scenario, which eliminates the pre-bond/intermediate test and uses the simulation adjusted die yield; and the partially functional case, which includes both non-testing and the hierarchical yield. The tolerant threshold is set to 1 and a two layer face-to-back design is used. The basic configuration is that the defect cluster radius is fixed as $10mm$, the wafer defect rate as 0.5 defects/cm^2, α as 0.5 in GNB model, and the die area as $50mm^2$ with four same size units per die. Then a case study on the multi-core processor design is elaborated and system costs with different design configurations are analyzed.

2.4.1 Impacts of Design Target

In this section, we vary the die area and the unit area to demonstrate the impact of design target on the final system cost. There are three variables in the design target: die area, unit area, and unit count. For each experiment, we fix one variable and change the values of other two. During the first experiment, the die area is changed while the number of units per die is kept constant. The results of the system cost are shown in Figure 2.6. Obviously, the larger the die size, the higher the total die cost. The increasing rates of both basic and non-testing case are the same, however, the case of partially functional dies has slower growing rate due to the increased partial yield when the unit size is large. Therefore, when the die size and unit size are large, using the partially functional die can help bias the cost.

Next, we fix the die area and change the unit area and unit arrangement. The

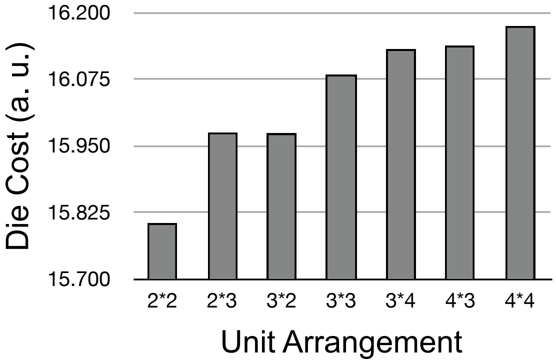

Figure 2.7. Cost variations with different unit area when the die area is fixed.

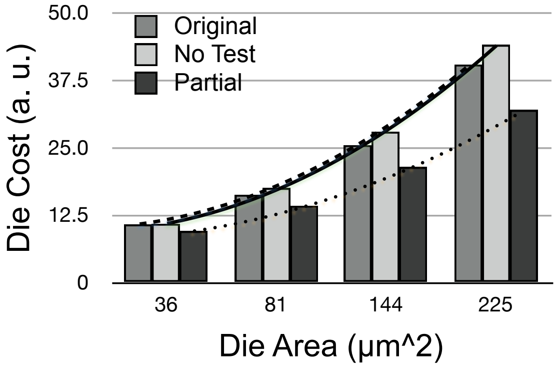

Figure 2.8. Cost variations with different die sizes/unit counts when the unit area is fixed.

cost with partially functional die is shown in Figure 2.7. The basic and non-testing costs are not listed because the die area is not changed in this experiment, leaving the partial yield the only variable. More number of units per die means smaller area per unit. When the number of units per die is fixed, the cost variation between different arrangements is small, as shown in bar 2×3 (representing the unit row and unit column) and 3×2. However, when the number of units per die increases from 4 (2×2) to 9 (3×3), the system cost are significantly increased. When the tolerant threshold and defect cluster degree are set, smaller unit means lower partial yield, because more units may be affected by defects.

In the first experiment, we assume the unit area increases as the die size increases, whereas the unit count per die keeps constant. However, to improve the area utilization, designers are tend to use small unit for packing more units inside one die. Therefore, we conduct another experiment to evaluate the system cost

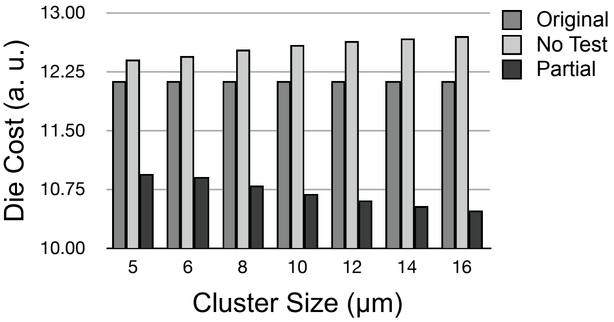

Figure 2.9. Cost variations with different defect cluster radius.

with increased die area when the unit size is fixed (i.e., the unit count increases). The solid line, dash line, and dotted line in Figure 2.8 show the cost increasing trend for the basic, non-testing, and partially functional scenarios, respectively. Compared with the results in Figure 2.6, even this design has lower partial yield due to smaller unit size compared to the first experiment, the trends are similar because the die yield is the dominant factor in cost estimation.

2.4.2 Impacts of Cost Parameters and Defect Clustering Degree

The partial yield is dependent on the ratio between unit size and defect cluster size. Therefore, we fix the unit size and vary the defect cluster radius to evaluate the system cost with results shown in Figure 2.9. As the cluster size becomes larger, meaning the defects are more random, the non-testing costs are increasing because more dies are defective and the die yield is low. In contrast, the costs with partial dies are decreasing due to the increased partial yield.

Intuitively, the cost parameters influence the final system cost. In this experiment, the test cost remains unchanged and the bonding cost and wafer cost are changed to study how the cost parameters change the system cost. We reduce the bonding costs to 30%, 50%, 65%, and 80% of its original value, respectively, and the wafer costs are reduced to its 50%, 60%, 70%, and 80%, respectively. The results in Figure 2.10 show that when the bonding cost is reduced to its 30% and 50%, the non-testing scenario is more cost-efficient. As the wafer cost reduces, the non-testing scenario is always overweight the basic cases because of the reduced die cost.

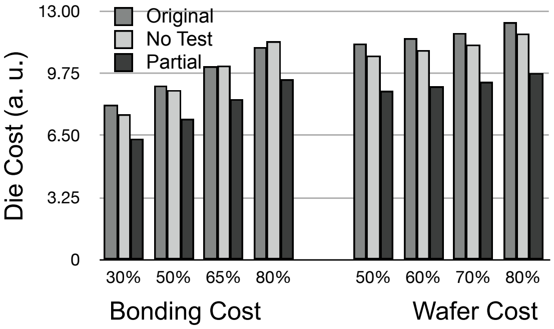

Figure 2.10. The impact of reducing wafer/bonding costs on integration costs.

Table 2.1. Multi-core processor design system cost comparison

Design	Original Cost	No Test Cost	Partial Cost
Small Design	10.77	11.26	9.74
Medium Design	9.18	8.98	8.18
Large Design	8.47	7.97	7.48

2.4.3 System Cost Analysis on Multicore Design

We use a homogeneous multi-core processor design as a case study and evaluate the SPARC-like core area using McPAT [112]. Assume a design with a total 16-issue width is needed. Areas for 2-, 4-, and 8-issue width cores with private L2 cache are $9.67mm^2$, $10.31mm^2$, and $12.53mm^2$, respectively. Assume a 2-layer 3D design is selected, there are three design options: four 2-issue width cores per layer (small core), two 4-issue width cores per layer (medium core), and one 8-issue width core per layer (large core). The cost results are shown in Table 2.1. As expected, when we use the large core, the die area is the smallest, therefore, the cost is the lowest due to higher die yield. However, the partial yield is the same as die yield in this case, there is no significant cost benefit from the partially functional die. Moreover, when one unit is defective in this layer, which means the whole large core is failed, the profit margin becomes negligible. Note that because medium core and large core designs have smaller die area, the die yield is therefore high enough for cost savings from the test elimination. The cost difference between the small core design and the medium core design is less than 3 units. If the marketing price of dies losing one small core is 3 more unit expensive than that of losing one medium core, then the small core design is more cost-efficient.

2.5 Summary

3D integration provides numerous benefits for future computer architecture designs. However, testing challenges and high integration cost hinder the commercial adoption of 3D designs. From the cost analysis with defect clustering, we find that test elimination can potentially reduce the system cost. Therefore, a cost benefit analysis framework is built to provide cost-efficient design options with given design targets. The corresponding testing elimination and partially functional die analyses are conduct under the consideration of defect clustering effects. Simulation studies show the system cost variation under different design and cost parameters, and the corresponding design guidelines are given.

Chapter 3

Metal Layer Reduction for Cost Optimization

In Chapter 2, the testing cost is analyzed and the possibility of test elimination for cost saving is shown. In the cost model, there are other factors that can influence the final cost. In this chapter, the potential cost saving is coming from the wafer cost with the emphasize on the mask cost[1]. Contemporary complex circuit designs require more metal layers, making the percentage of mask cost continuously increases. For example, the process steps are 462 with mask cost of 2.974M/set in 32nm 9 metal layer technology node. When the metal layer number increases to 11, the process steps increase to 498 and mask cost is 3.212M/set. Reducing metal layer can cut down the process steps and mask cost. It is impossible to reduce metal layer in 2D design when we want to maintain small die size and feasible routability. However, with the benefits provided by 3D stacking, metal layer reduction becomes possible.

In this chapter, we built a block granularity cost-driven 3D design exploration flow to find the cost-efficient design. This flow finds the best design by balancing the chip area (placement density) and routability (metal layers). In practical design, designers are requested to input the area utilization to indicate placement density. But the input area utilization may not be optimal. In this flow, cost efficient placement density is given with metal layer reduction. Both TSV-based

---

[1] This work is published as "Cost-drive 3D design optimization with metal layer reduction" on ISQED 2013.

3D and interposer-based 3D designs are considered in this analysis.

3.1 Routability and Cost Models

In order to reduce metal layer in 3D design, we need to estimate the minimum metal layer requirement that can guarantee the routability given a certain design gate count. Besides the interconnect model, 3D system cost models for TSV and interposer based 3D ICs are necessary for cost analysis. In this section, the related interconnect model and 3D cost model are introduced. In the interconnect model, we describe how to estimate the die area and minimum metal layers for feasible routing, when only limited information are available. The second part of this section introduces a comprehensive 3D TSV bonding cost model which considers the area overhead and yield impact of TSVs and silicon-based interposer cost model. These models will facilitate the cost efficient 3D design flow in Section 3.2.

3.1.1 Routability Models

The die area, which is defined as the area occupied by the transistors and the interconnects, is closely related to the gate count in designs [113]. However, in practical circuit designs, gates are not placed tightly with other, which introduces an area utilization rate to indicate the placement density. The utilization rate together with total gate size determine the die area in floorplan. So the die area can be estimated as a function of the gate counts and area utilization rate:

$$A_{die} = \frac{N_g A_g}{A_{util}} \tag{3.1}$$

where N_g is the number of gates in the design, A_g is an empirical parameters that represents the relation between single gate area and feature size, A_{util} is the area utilization percentage. In our work, A_g is assumed to $3125\lambda^2$, and λ is half of the feature size.

The required metal layer for feasible routing is depended on the interconnect complexity, however, the detailed connection information is unavailable at the early design stage. In this work, we mainly analysis the relation between routing demand and routing resources to determine the required metal layer number. For the routing demand, a wire length distribution model based on Rent's rule is used to perform the estimation [114]. The wire length distribution function with respect to the interconnect length l is as follows:

Region I: $1 \leq l \leq \sqrt{N_g}$

$$i(l) = \frac{\alpha k}{2}\Gamma\left(\frac{l^3}{3} - 2\sqrt{N_g}l^2 + 2N_g l\right) l^{2p-4} \tag{3.2}$$

Region II: $\sqrt{N_g} \leq l < 2\sqrt{N_g}$

$$i(l) = \frac{\alpha k}{6}\Gamma\left(2\sqrt{N_g} - l\right)^3 l^{2p-4} \tag{3.3}$$

where k is Rent's coefficient, which means the average number of pins per block and p is Rent's exponent [115] within the range of 0.4 to 0.9 [116]. $\alpha = \frac{f.o.}{f.o.+1}$ is related to the average fanout of a gate to represent the proportion of on-chip sink terminals.

and Γ is given as follows:

$$\Gamma = \frac{2N_g\left(1 - N_g^{p-1}\right)}{\left(-N_g^p \frac{1+2p-2^{2p-1}}{p(2p-1)(p-1)(2p-3)} - \frac{1}{6p} + \frac{2\sqrt{N_g}}{2p-1} - \frac{N_g}{p-1}\right)} \tag{3.4}$$

For any given interconnect length, the accumulative number of interconnects $I(l_i)$ means the number of interconnects with length smaller or equal to l_i can be derived from the cumulative integral of the wire length distribution function $i(l)$. The accumulative interconnect length $L(l)$ is given as the first-order moment of $i(l)$ which can represent the routing demand.

The routing supply is related to the routable area and wire pitches. The available signal routing resources is significantly smaller than the total track length on all metal layers because of the routing efficiency, impact of the vias, and resources occupied by power, ground and clock distribution [117]. The available signal routing resources for each metal layer is given as follows:

$$K_i = \frac{A_{die}\eta_i - 2A_v\left(I(l_{max}) - I(l_i)\right)}{\omega_i} \tag{3.5}$$

where A_{die} is the die area given in equation 3.1, η_i is the metal layer utilization considering routing efficiency and non-signal routing resources occupancy, A_v and ω_i are the via area and wire pitch on metal layer i, l_max and l_i are the maximum interconnect length on the whole chip and one metal layer.

By using this routing resources analysis, we are assuming the shorter interconnects are routed first on lower metal layers. The routing process starts from metal layer 1, and moves up the higher metal layer only when the bottom layers are fully utilized. And for each metal layer, the total routed wire length cannot exceeds the available routing resources. The required metal layer number can be derived from repeating this routing process until all the interconnects are properly assigned to metal layers.

3.1.2 3D Cost Model

The cost models are different in TSV based and interposer based designs. In our work, the interposer is treated as regular silicon device, such that only the wafer cost is considered. For 3D TSV based structures, the cost contains two parts, wafer cost and bonding cost. Wafer cost captures the wafer cost with related die cost and die yield, similar to traditional 2D design. Bonding cost models the bonding cost and bonding yield, which is unique in 3D designs.

Wafer Cost Model. In the wafer cost model part, the most important factor is the die area. The die area estimation of TSV bonding structure and interposer structure are different. TSV based 3D structure introduces area overhead, because the silicon area where TSVs are built cannot be utilized. Normally, the diameter of TSVs is large compared to the device feature size and it can range from $1\mu m$ to $10\mu m$ [14]. The area overhead caused by TSVs can be estimated through the formulation $A_{3D} = A_{die} + N_{TSV/die} * A_{TSV}$ [96]. $N_{TSV/die}$ refers to the number of TSVs on each die, A_{TSV} is the area of TSV and it can be calculated when given the TSV pitch, and A_{3D} is the final die area of one 3D stacking die.

For interposer stacking, the vertical interconnects have no impact on chip are because the TSVs are fabricated inside interposer. So the die area in interposer cases can be estimated directly from Equation 3.1. Given the die area and wafer diameter, the wafer utilization in terms of number of dies per wafer can be formulated.

Besides the number of dies per wafer, the die yield also influences the cost. Given the wafer defect density and die area, the die yield is formulated as follows:

$$Y_{die} = Y_{wafer} * \frac{1 - e^{-2A_{die}D_0}}{2A_{die}D_0} \tag{3.6}$$

where D_0 is the defect density of the wafer, Y_{die} and Y_{wafer} are the die yield and wafer yield respectively.

Since 3D integration enables heterogeneous stacking, it is not necessary to have all the stacking dies with the same chip size. For the whole 3D stacking, the overall die yield can be calculated by multiplying the individual yield of the dies in the stacking.

3D Bonding Cost. In order to build 3D stacking, extra fabrication steps are needed, such as wafer thinning, TSV forming and die bonding. Among these three process steps, wafer thinning decreases the wafer yield, TSV forming increases the process cost, and die bonding influences both the process cost and stacking yield. The total stacking yield is function of the number of TSVs (N_{TSV}) and the yield of single TSV (Y_{TSV}), and it can be calculated by $Y_S = Y_{bonding} * Y_{TSV}^{N_{TSV}}$ [97]. The number of TSVs in a design can be estimated as follows at the early design stage:

$$\begin{aligned} N_{TSV} &= \alpha k_{1,2}(B_1 + B_2)(1 - (B_1 + B_2)^{p_{1,2}-1}) \\ &\quad -\alpha k_1 B_1 \left(1 - B_1^{p_1 - 1}\right) - \alpha k_2 B_2 \left(1 - B_2^{p_2 - 1}\right) \end{aligned} \tag{3.7}$$

where B_1 and B_2 are the number of blocks in two tiers, $k_{1,2}$ and $p_{1,2}$ are the equivalent Rent's coefficient and exponent.

Overall 3D Cost Model. The 3D interposer stacking cost only contains the wafer cost model. And in our work, we consider the case that is shown in Figure 3.1(b) where two known good dies stack on both sides of an interposer. In this situation, the overall cost of one interposer stacking is given as follows:

$$C_{IPstacking} = \sum_{i=1}^{N}(C_{die_i} + C_{KGDtest}) + \sum_{i=1}^{N/2} C_{interposer} \tag{3.8}$$

where N is the number of tiers in the 3D design, C_{die_i}, $C_{interposer}$ are the cost of each tier in the stacking and interposer wafer cost, respectively, $C_{KGDtest}$ is the

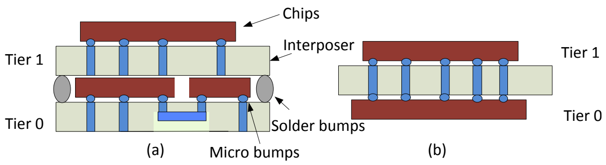

Figure 3.1. An example of 3D interposer stacking. (a). Stacking chips are on one side of interposer; (b). Stacking chips are on both sides of interposer.

cost of testing for each die.

The 3D TSV bonding cost combines the wafer cost model and bonding cost model. For die-to-wafer stacking, the individual dies are cut from wafer first and then stacked on the bottom wafer after testing. Therefore, only the known-good-dies are put into the stacking. On the other hand, for wafer-to-wafer stacking, wafers are bonded together before testing. Consequently, die yield and stacking yield have impacts on the final fabrication yield. In this analysis, we only consider die-to-wafer stacking, the cost can be calculated from:

$$C_{D2W} = \frac{\frac{\sum_{i=1}^{N}(C_{die_i}+C_{KGDtest})}{Y_{die_i}} + (N-1)C_{bonding}}{Y_S^{(N-1)}} \tag{3.9}$$

where $C_{bonding}$ is the stacking cost, it captures the wafer thinning, TSV forming and bonding cost.

3.2 Cost Efficient 3D Design Exploration Flow

Based on the interconnect model and 3D cost model, 3D cost efficient design exploration is performed in block granularity to reduce system cost with metal layer reduction. In this section, we introduce our 3D design flow with metal layer reduction in detail.

The flow takes the gate counts from synthesized design as input and outputs the cost efficient 2D, TSV based and interposer based 3D designs. Figure 3.2 shows the design space exploration flow. In this work, the 3D partitioning is in block granularity and each block contains arbitrary number of gates according to the block functionality.

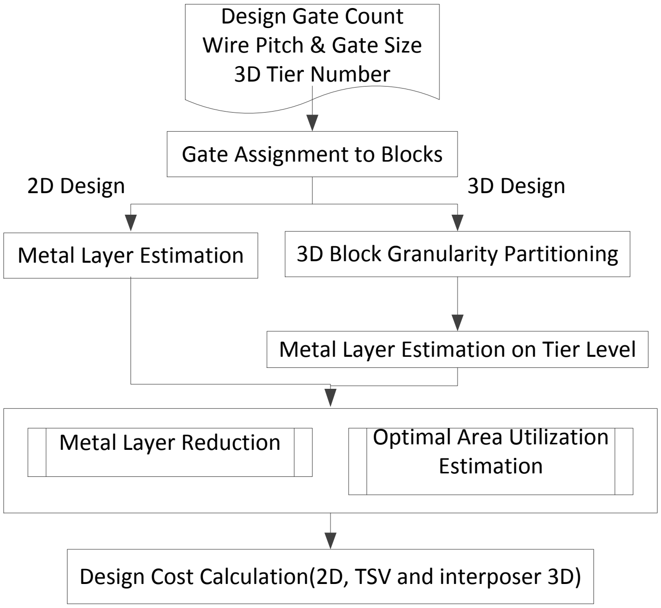
Design Gate Count
Wire Pitch & Gate Size
3D Tier Number
Gate Assignment to Blocks
2D Design
3D Design
Metal Layer Estimation
3D Block Granularity Partitioning
Metal Layer Estimation on Tier Level
Metal Layer Reduction
Optimal Area Utilization Estimation
Design Cost Calculation(2D, TSV and interposer 3D)

Figure 3.2. Cost efficient design exploration flow in 2D, 3D TSV bonding and 3D interposer stacking integrating metal layer reduction technique.

After 3D partitioning, the traditional 2D design, TSV and interposer based 3D designs are generated with estimated chip size using default area utilization. The default metal layers for feasible routing are calculated using the interconnect model introduced in Section 3.1. Given the estimated chip area and default metal layers, fabrication cost of designs are calculated. The most cost-efficient designs are selected as suggested solution.

In this flow, the most significant parts are metal layer estimation and corresponding utilization estimation. For, metal layer estimation, the routability model in Section 3.1 is used. However, the l_i, which is the maximum interconnect segment length that can be routed on metal layer i, is constrained by the routing resources. The routing resources on the other hand is bounded by the via area, which depends on l_i. For example, on metal layer 1, if the maximum length that can be routed is l_1, then the connections that are longer than l_1 need to be put on higher metal layer, which results in $I_{max} - I_1$ number of vias. The bigger l_1 value is, the less area occupied by vias. But the bigger l_1 needs more routing resources determined by available chip area. Finding the balanced l_i for each metal layer is the key step in metal layer estimation. Searching all the possible combinations in sequence is extremely time consuming. In this work, we are using binary search algorithm as shown in 1. The algorithm takes the chip area and wire pitches as input and gives the metal layer number as output.

Algorithm 1 Outline of the binary search algorithm for metal layer requirement estimation.

```
Metal Layer Estimation (chip area, wire pitch)
{initialize layer count, $l_{max}$, $L(l_{max})$ and $I(l_{max})$}
layer count = 0;
while $l_i \leq l_{max}$ do
  $l_i$ = (upbound+lowbound)/2;
  repeat
    Calculated $k_i$ from $l_i$;
    Calculate $L(l_i)$ from $l_i$;
    if $L(l_i) - L(l_{i-1}) < k_i$ then
      Update the lowbound;
    else
      Update the upbound;
    end if
  until lowbound > upbound
  Finish searching on metal layer i; i++;
  Record $L(l_i)$ and $l_i$;
  {find the balanced $l_i$}
end while
layer count = i;
return  Layer count;
```

Area utilization is the major factor that influences the fabrication cost. First, die yield is exponentially related to the die area which is calculated from area utilization. It means larger chip area results in significant higher cost. Second, the required metal layer is determined by the area utilization which determines the routing resource. Increased area utilization results in denser placement and more metal layers, implying higher mask cost. In our work, we consider the maximum area utilization that one design can achieve without sacrificing routability. **Optimal area utilization** means the maximum utilization with default metal layers and **maximum area utilization** is the maximum area utilization after one metal layer reduction. The design exploration flow finds the cost efficient design through calculating the optimal and maximum area utilization. In practical design, CAD tools usually require designers to input the area utilization and perform placement and routing accordingly. However, sometimes the utilization is pessimistically estimated resulting in large unused chip area. So finding the cost efficient area utilization can help reduce the die area and cost with feasible routing.

The same binary search algorithm is used for area utilization exploration. The optimal area utilization is searched between the default utilization rate and maximum utilization threshold. The maximum utilization rate without causing additional metal layers is the output. The maximum utilization rate after metal layer reduction should be between the minimum utilization threshold and default utilization.

The metal layer and area utilization estimations are suitable for general propose logic designs. However, the estimation is inaccurate for regular pattern designs, such as cache and memory. In these regular designs, the routing complexity is not related to the number of gates and the metal layers are fixed once design pattern is known.

3.3 Experiment Results

In this section, the results from cost efficient design exploration are shown. We use 45nm technology node for the experiments. The gate size and wire pitches information are extracted from NanGate FreePDK45 Generic Open Cell Library [118]. The parameters used in the experiments are listed in Table 3.1. During the design exploration process, the area utilization adjustment is within the range of 20% to keep the design practical. The cost related parameters in this experiment are from IC cost model [119]. For device layer we use 300mm TSMC dual gate CMOS logic process technology and for interposer layer we use 300mm UMC interposer process technology. 3D bonding method is face-to-back bonding and die-to-wafer stacking.

3.3.1 Cost Analysis without Optimization

The costs of 2D, 3D TSV, and 3D interposer implementations of each design with default area utilization are calculated. In this section, the cost of 2 layer 3D partitioned design without optimization is shown in Figure 3.3.

The product cost rises with increased gate count as expected. When the gate

Table 3.1. Parameters used in the experiments

Parameter	Value
Average gates per block	50
Average gate fanout	2
Rent's coefficient (k)	4
Rent's exponent (p)	0.7
TSV area	$1\mu m^2$
Default area utilization	70%
Routing efficiency	10%

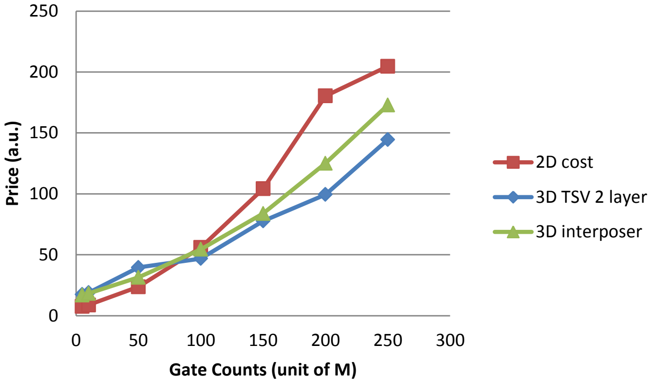

Figure 3.3. Cost comparison between design implementation on 2D, 2 layer 3D TSV, and 2 layer 3D interposer designs.

count is smaller than 100M, the 2D implementation has lower cost than all 3D cases. However, as the gate count increased, 3D designs show the advantages of smaller footprint and less routing complexity over 2D designs. In the proposed flow, we only consider 2 layer partitioning for 3D interposer stacking, and the interposer area is determined by the maximum chip size. When the gate count increased, the cost of interposer is higher than 3D TSV bonding, because of the lower die yield and higher wafer cost of larger interposer area. In previous work [96], the gate count of 3D enabling point is less than 100M, which is smaller than our results because the previous process technology is 65nm. As transistor size continuously scaling, the chip size is reduced, and therefore amortized cost per chip is lower. 3D stacking designs are more cost efficient for large designs.

Besides 2 layers partitioning, 3D designs costs with more layers are examined.

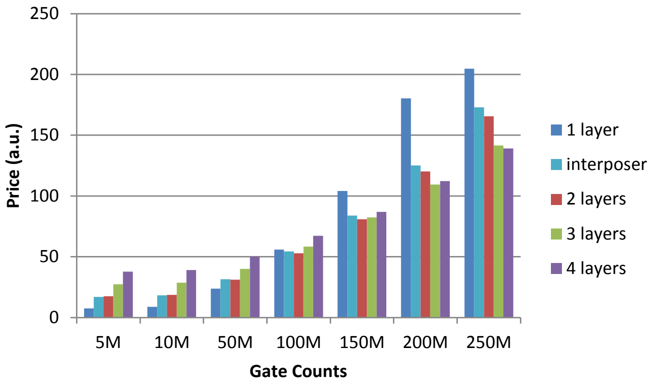

Figure 3.4. Product costs of 2, 3, and 4 layers TSV based 3D designs, 2D and interposer based design costs are shown for comparison.

Table 3.2. Results of optimal designs after cost efficient optimization process for 2D, TSV and interposer based 3D.

Gate Count	**2D Design**			**3D TSV Bonding**			**Interposer Stacking**	
	metal layer	area util (%)	cost	tier num	area util (%)	cost	area util(%)	cost
5M	3	89	7.17	2	89	17.06	89	16.51
10M	3	81	8.18	2	89	17.93	89	17.44
50M	5	79	18.34	2	79	27.26	79	27.47
100M	6	78	38.81	2	79	41.88	79	43.18
150M	6	72	78.37	2	73	65.72	73	68.77
200M	7	71	131.85	3	75	87.97	78	89.45
250M	8	70	204.62	4	76	111.52	75	127.01

The costs are shown in Figure 3.4. In 3D TSV designs, more layers do not necessarily mean higher cost. From the results, for larger designs with gate count more than 150M, we can gain cost saving by partitioning the design into more tiers. When the design is large, the additional cost introduced by TSV bonding is compensated by smaller footprint since the die yield is exponentially related to the chip size.

3.3.2 Cost Analysis with Optimization

In this section, the results after cost efficient optimization are shown. For each design (both 2D and 3D cases), the optimal utilization and maximum utilization are obtained to calculate corresponding design costs. The results are shown in Table 3.2. In the table, the shown utilization is the most cost efficient utilization.

Table 3.3. Cost Savings of 2D, TSV and interposer based 3D designs for optimal area utilization with default metal layers.

Gate Count	50M		100M		150M		200M		250M	
2D Savings	5.34	22.54%	17.09	30.58%	25.83	24.79%	48.53	26.9%	0	0%
3D TSV	3.86	12.41%	11.04	20.87%	15.06	18.65%	21.48	19.62%	27.52	19.79%
3D Interposer	3.96	12.59%	11.25	20.68%	15.03	17.94%	33.65	28.50%	45.82	26.51%

Among all the designs, the most cost efficient designs are obtained when the area utilization rate is higher than the default utilization, which means the area is dominant factor of cost. In 2D designs, the optimal utilization decreases when the gate counts increases. The optimal utilization is achieved when all the metal layers are fully utilized for signal routing. For small designs, the top metal layer is usually with low utilization, only a small portion of interconnects is used. The large designs with higher gate counts require more routing resources causing high utilization even in high metal layers. It makes the large designs harder to shrink the chip area. In this results, 89% chip area utilization is the optimal utilization for 5M gate count designs, but for designs with 150M gate counts, the optimal utilization decreases to 72%.

After the design optimization, the costs of 2D and 3D designs are significantly reduced. Cost savings are shown in Table 3.3. 2D designs have the highest saving on average except the last design when the original design before optimization is already the best one. TSV and interposer based designs have significant cost saving after the optimization. For large designs, the benefits are higher because of larger flexibility in chip area.

Although chip area reduction is the major factor for cost saving, metal layer reduction also provides cost saving compared to baseline designs. The cost saving results from metal layer reduction is shown Table 3.4. In order to maintain feasible routing after one metal layer reduction, chip area is increased to provide additional routing resources. The area increment percentage depends on different designs. For example, designs with low utilized top metal layer can have metal layer reduction with very small area overhead. But for some designs, area needs to increase about 10% to enable one metal layer reduction, thus the cost saving is negative since cost overhead for larger chip area overweighs the cost saving from metal layer reduction. For 2D designs with 200M gate counts, when the area increases just

Table 3.4. Cost Savings of 2D, 3D TSV bonding, and 3D interposer stacking designs with metal layer reduction technique.

Gate Count	50M	100M	150M	200M	250M
2D Savings	3.01	10.19	6.04	38.57	-11.78
2 layer 3D TSV	-8.46	6.11	3.08	20.73	21.09
3 layer 3D TSV	-4.11	0.98	9.28	7.83	2.69
4 layer 3D TSV	-3.85	3.16	0.46	12.51	11.62
3D Interposer	1.43	5.94	2.16	20.18	19.68

1%, the required metal layer can be reduced. Thus the cost saving for the design is high. For 3 layer 3D TSV designs with 50M gate counts, in order to gain one metal layer reduction, the area needs to increase 20%, so no cost saving can be achieved from metal layer reduction. On average, the area increment is within 10% for designs to have one metal layer reduced in 2D and 3D cases.

3.4 Summary

In this chapter, a 3D design cost efficient exploration flow is proposed to find the cost efficient designs in 2D and 3D cases. Both TSV and interposer 3D structures are considered. The cost efficient designs is found with balanced area utilization and metal layers. The experiment results show that for both 2D and 3D cases, metal layer reduction and optimal area utilization exploration can achieve cost saving compared to baseline designs. We can achieve cost saving up to 19% for TSV-based 3D designs, and up to 26% for interposer-based 3D designs, respectively, compared to the baseline designs.

Chapter 4

Vertical Bandwidth Reconfigurable 3D NoCs Utilizing Redundant TSVs

The last decades we have witnessed a growing trend of packing many processing elements (such as cores, caches, etc.) in a single chip. Conventional multi-core design adopts a single bus with limited bandwidth as the communication backbone. Consequently, the bus bears enormous stress and even becomes the performance bottleneck due to frequent packet transmission, leading to recent many-core chips interconnected with sophisticated Network-on-Chip (NoC) [120]. Such systems have routers at every node, connected to neighbors through short links, while multiplexing packet flows at each router to provide high scalable bandwidth[1].

Therefore, the computing system performance can be further advanced by extending planar NoC with 3D integration technology for robust and high bandwidth intra/inter-layer communication [121]. Plenty of studies have explored the benefits of adopting 3D NoC in system designs with different topologies or architecture configurations [122, 123]. These studies demonstrate that 3D NoCs are capable of achieving higher throughput, lower latency, and lower energy dissipation with only slight area overhead.

In NoC designs, the channel width of each direction is predetermined at the design time. However, when running diverse applications on a many-core system, the runtime workloads pose sporadic stresses on the network, making the static

---

[1] This work is published as "Designing vertical bandwidth reconfigurable 3D NoCs for many core systems" on 3DIC 2015.

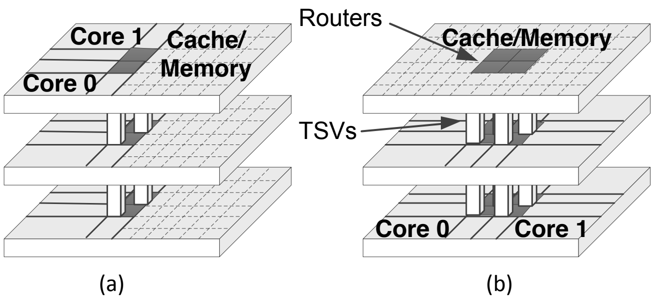

Figure 4.1. Overview of core-to-cache/memory 3D stacking. (a). Cores are allocated in all layers with part of cache/memory; (b). All cores are located in the same layers while cache/memory in others.

configuration inefficient [124, 125]. This observation is more prominent in 3D core-to-cache/memory stackings as shown in Figure 4.1, which are the most prevalent and practical 3D system designs. If cores and memory/caches are on different layers (Figure 4.1(b)), the percentage of the interlayer traffic is large [126]. Moreover, considering the large size of TSVs, it is infeasible to arbitrarily increase the channel width due to the power and area constraints. Assuming the pitch of TSV is 10μm, the total TSV area for a 128-bit channel is 12,800μm^2. In contrast, the total area of a 3D router is 84,669μm^2 based on our synthesized result, indicating a significant overhead of TSVs. Therefore, in a 3D system with severe interlayer communication congestion, it is undesirable to add additional TSVs for more vertical bandwidth. Instead, a more cost-efficient way is to dynamically allocate the available channel bandwidth under a fixed budget.

Fortunately, we reveal that there are available vertical connection resources which are not fully utilized. In 3D systems, redundant TSVs are usually used as the simplest yet effective remedy to guarantee signal integrity and improve system yield [127, 128]. However, many redundant TSVs that are statically allocated are not utilized during chip operation when the number of faulty TSVs is smaller than the number of redundant ones. Therefore, these spare redundant TSVs can be potentially leveraged to increase vertical bandwidth for instant throughput improvement. In this chapter, we propose a 3D NoC design with reconfigurable vertical channel width, which enables existing redundant TSVs to be flexibly shared by nearby NoC routers. When the additional channel width is not necessary, spare TSVs will be disconnected from the router. In addition, the vertical interlayer traffic are dynamically monitored, and the spare TSVs are allocated to the bandwidth hungry routers accordingly. To support this mechanism, we designed an additional link (AL) allocator which incurs negligible hardware overhead. Note that a single link denotes a bundle of wires for signal transmission in one direction.

4.1 Preliminaries and Related Work

In this section, we first introduce our target 3D system and the background of TSV redundancy. Then we present the motivation of designing vertical bandwidth reconfigurable NoCs.

4.1.1 3D Multicore Processor Designs

In current 3D IC designs, the most attractive and prevalent stacking style is the core-to-cache/memory bonding [4, 123]. By stacking the data storage component close to the logic computing component, the cache/memory access latency can be greatly reduced. Moreover, the issue of limited pin count can be solved after migrating the memory to the third dimension, resulting in higher bandwidth.

Figure 4.1 shows an example of 3 layer core-to-cache/memory stacking with two potential core-to-memory partitioning schemes. NoC routers are placed near each core and memory controller to provide high performance connection. In the Figure 4.1(a), each layer contains several cores, and parts of the data storage are placed close to the core on the same tier. To access the data from other layers, TSVs are used as the vertical channel to connect routers. Figure 4.1(b) shows an alternative partitioning scheme. All the cores are located on the same layer, while all the cache/memory modules are spread out on the rest tiers. Under such circumstance, all data access requests from cores need to go through vertical channels.

Due to the elevated thermal dissipation in 3D designs, the heat dissipation of cores on the layers that are far from the heat sink may cause severe problems. Consequently, the design using partitioning method in Figure 4.1(a) should be carefully handled to balance the temperature and performance. In contrast, the design in Figure 4.1(b) has less potential thermal problem thanks to the direction contact between cores and the heat sink. However, this design has higher vertical bandwidth demands since the memory request cannot be fulfilled by 2D connections. Therefore, the NoC design can provide predictable and high performance communication under certain bandwidth limitation compared to bus connection. We focus on the second design (Figure 4.1(b)) because it is a more favorable design which achieves better thermal behavior because of the short distance to the heat sink. Also, it poses more challenges on 3D NoC designs due to the frequent vertical core-to-memory accesses. However, our design is also applicable to other 3D systems.

4.1.2 TSV Redundancy

TSV is one of the key enabling technologies in 3D integrations, which is fabricated by forming a hole through the silicon and filling the hole with conductor materials. Short length, high densities, and high compatibility with standard CMOS process are three major reasons making TSV attractive in the 3D technology. However, TSVs suffer from low yield due to the manufacturing limitation compared to 2D wires [129, 130].

Misalignments and random open defects cause the failure of TSVs [129]. In order to guarantee the signal integrity, building redundant TSVs is a common mechanism for fault tolerance. A straightforward method of building redundant TSVs is to double the TSV count for each signal. However, this method will incur severe area overhead. Alternatively, certain ratio of redundancy is applied in design time. For example, in 3D DRAM design [128], every 4 TSVs are allocated as a group, while 2 additional TSVs are attached to this group for fault tolerance. A switch box is used to select any four functional TSVs from these six for the signal transmission. To further reduce the area overhead, several previous studies [110, 127, 129, 130] have proposed mechanisms to efficiently allocate redundant TSVs and perform self-repair.

Even though these work can enhance the chip yield, certain number of redundant TSVs (e.g., redundancy ratio is 1:4 in [127]) should be allocated statically without knowing the actual failure in design time. Subsequently, some of the allocated redundant TSVs may be unused, resulting in resource wasting. For instance, since the single TSV failure rate is 10^{-5} to 10^{-4} [127] based on current packaging technology, when the vertical signal count is 128, the probability that all TSVs are functional is 95% to 99.49%. Nevertheless, the redundant TSVs can not be eliminated as any TSV failure would lead to performance loss. Other work considers the fault tolerant scheme without redundant TSVs [131]. Nevertheless, the signal needs to be serialized if the available TSVs count is smaller than the signal width, which leads to performance loss.

4.1.3 Motivation and Related Work

In NoC designs, the channel bandwidth has significant impact on the packet latency. Specifically, a larger network bandwidth is capable of reducing the transmission delay over the channel. To constrain the area overhead and routing complexity, one NoC packet is usually partitioned into several flits, and the channel width in each direction is designed to accommodate one flit. For each NoC cycle, at most one flit is transmitted per direction. The whole packet transmission is completed only when all flits are received by the destination.

The packet latency can be reduced if additional channel is available to allow simultaneous transfer of multiple flits. Figure 4.2 shows a timing diagram of a virtual-channel router containing four stages: routing computation (RC), virtual channel allocation (VA), switch allocation (SA), and switch traversal (ST). Assuming a scenario that packet A is from input 1 while packet B is from input 2, and they are requesting the same output port. In Figure 4.2(a), at cycle 3, since only one flit is allowed to access the output port, and the head flit of packet A is granted the access, the head flit of packet B has to stall until cycle 4. If an additional link is available before cycle 3 in Figure 4.2(b), then both head flits from packet A and B have accesses to the output port. The transmission of head flit in B is advanced from cycle 5 to cycle 4.

To relieve the intermittent traffic congestion, various bandwidth reconfigurable NoC architectures are proposed [124,125,132]. Lan et al. [124] proposed to use bidirectional links instead of traditional unidirectional links. Fine-grained bandwidth adaptive NoC and bandwidth-adaptive oblivious routing [125, 132] are proposed to further increase the channel utilization based on the bidirectional link architec-

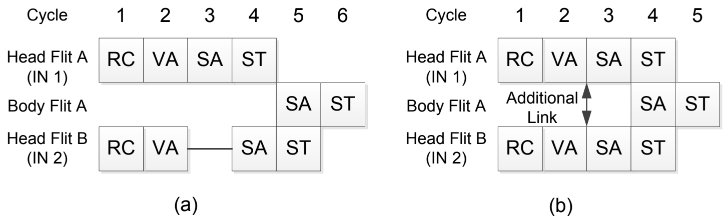

Figure 4.2. Timing diagram of the packet latency reduction with additional channel width. (a) Original design with fixed channel width; (b) Designs with additional output channel width.

ture. Their designs increase bandwidth in one direction at the cost of sacrificing bandwidth in another direction. In our work, we use the spare TSVs as additional channel resources to increase the channel bandwidth. In addition, we have negligible modification to the original router architecture and routing algorithm, minimizing the design overhead.

4.2 Reconfigurable Vertical Link Design

In this section, the baseline router architecture and our spare TSVs allocation strategy are introduced. Other design issues, such as router placement and routing algorithms, are also discussed.

4.2.1 Basic Router Architecture

In a typical NoC design, a router usually consists of input buffers, switches, computing logics, and control logics. Various router architectures exist in current designs. Without loss of generality, we use the classical five-stage virtual-channel router design [133] as the baseline router architecture. Nevertheless, our mechanism also applies to other router architectures, such as speculative routers with shorter pipelines.

As shown in Figure 4.3(a), the virtual-channel router contains six functional blocks: input buffer, switch, output unit, routing computation, virtual channel allocator, and switch allocator. They can be categorized into two parts: the datapath which contains the first three components and the control panel with the later three. The datapath handles the storage and movement of network data while the

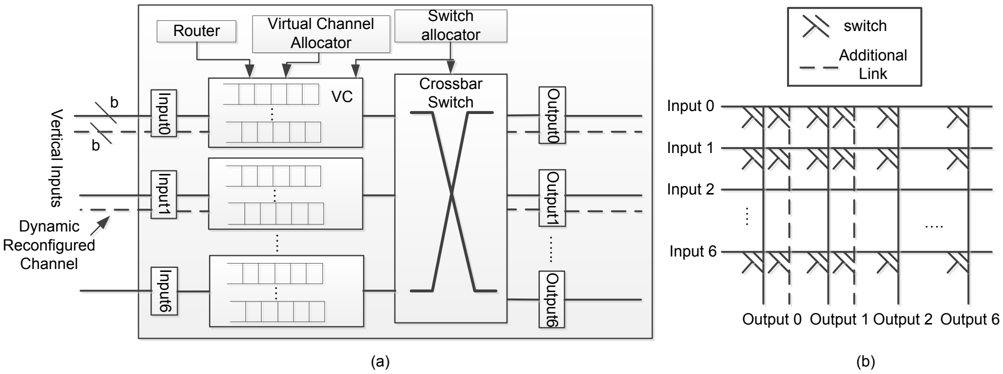
Router
Virtual Channel Allocator
Switch allocator
VC
Crossbar Switch
Input0
Input1
Input6
Output0
Output1
Output6
b
b
Vertical Inputs
Dynamic Reconfigured Channel
(a)
switch
Additional Link
Input 0
Input 1
Input 2
Input 6
Output 0
Output 1
Output 2
Output 6
(b)

Figure 4.3. Virtual-channel based router structure and the corresponding crossbar. (a). Virtual-channel based router architecture; (b). The crossbar design supporting additional vertical links.

control panel is responsible for routing computing and resource allocation.

Usually in the signal transmission of NoCs, one packet is split into several flits, which is the basic unit of network data transmission. A header flit will trigger the routing and virtual channel allocation. When the virtual channel allocation succeeds, the designated virtual channel is reserved for the rest flits in that packet. Therefore, the upcoming flits only require switch allocation and switch traversal. Switch allocation records output requests and grants the access of outputs to inputs based on the allocation strategy (e.g. round-robin). Every cycle, only one virtual channel in a router can get access to one output. If output requests are conflict, inputs without grant have to wait for the next cycle. After switch allocation, the switch allocator will configure the crossbar to conduct flit traversals.

4.2.2 Proposed Architecture Modification

As illustrated above, to avoid switch conflicts, each output can forward data from only one virtual channel per cycle. Therefore, in a congested network, as in the core-to-cache/memory configuration, the vertical output request overflows, resulting in large packet delay. For example, if all six inputs in one router are all requesting the same vertical output. Then at least six cycles are needed to finish the switch allocation of these requests.

In our work, we propose to use the spare TSVs to enhance the vertical channel bandwidth. Modifications to the basic router design are essential. In additional to the physical link connection, the switch allocator and crossbar are two important components that should go through architectural modification. Moreover, to reduce the area overhead of TSVs, we adopt the assumption that the TSV redundancy ratio is 1:4 [127], which means that every four routers share one group of redundant TSVs. Due to the resource sharing between routers, we propose a new component called additional link (AL) allocator to grant access of spare TSVs.

4.2.2.1 Switch Allocator Design

When a router has additional links, more flits can go through the vertical channel every cycle. However, the link width between each input to the crossbar is fixed as one flit size, forwarding one flit to the crossbar every cycle. Prior studies show that network performance can be improved with input speedup, which increases the link width between each input and the crossbar at the cost of additional area and energy [133]. In our mechanism, the switch allocator simply selects two inputs using the original allocation algorithm instead of two virtual channels in one input (as in input speedup). Moreover, the input speedup mechanism is orthogonal to our work, and it can be applied in our design for further performance improvement.

4.2.2.2 Crossbar Design

Even though spare TSVs are allocated dynamically in run-time, modifications to the crossbar are necessary in design time to enable connection between inputs to spare TSVs. Figure 4.3(b) depicts the crossbar design. Two additional links are added to vertical output ports: one for communication to the upper layer and one for the lower layer. Switches in the crossbar control the connection between inputs and outputs. When spare TSVs are not allocated to that router, the switch allocator turns off switches for additional links.

4.2.2.3 Additional Link Arbitrator

There are two assumptions with the additional link allocation. First, the physical channels use unidirectional links to reduce the control complexity of bidirectional links [124]. Second, the allocation of two links with opposite directions can be decoupled, which means the additional links to the upper layer and lower layer can be assigned to different routers.

The major challenge of designing AL arbitrator is to identify the congestion level of different routers. Intuitively, we can use the total vertical request count as the congestion indicator. However, this method fails to capture the real congestion status of a router. Considering the case that there are three vertical output requests in router A and two requests in router B, and the arbitration grants router A the permission since it has more requests. Nevertheless, all of the requests in router A come from the same input port, although in different virtual channels. While in router B, the two requests are from different inputs. According to the switch allocation mechanism, only one request per input can be fulfilled each cycle. Therefore, the performance of router A will not be improved even though it acquires spare TSVs. In this case, granting the access to router B is an efficient choice. In this sense, the congestion indicator should be in the granularity of input ports instead of the virtual channels.

After each flit cycle, each router will update its congestion level to the AL arbitrator. Then the AL arbitrator selects the router with the heaviest congestion and grants the permission before the next switch allocation cycle.

However, the increase of vertical bandwidth will pose stress to the downstream router (the next connected router) since it may receive more flits than it can process right away. We further propose an arbitration strategy to tackle this problem. Specifically, in addition to the vertical request count (VRC) to the downstream router as illustrated above, we obtain the buffer occupancy count (BOC) of the input port for this router, then the difference between VRC and BOC accurately indicates the necessity for additional bandwidth. As a result, the AL arbitrator grants access of spare TSVs to the router with the highest $VRC - BOC$ value.

4.2.3 Design Issues

Besides the architecture modification, there are some other important issues to be addressed in order to achieve a high performance 3D NoC design. In this section, we discuss the design issues including router placement, flow control, the availability of spare TSVs, and the routing algorithm.

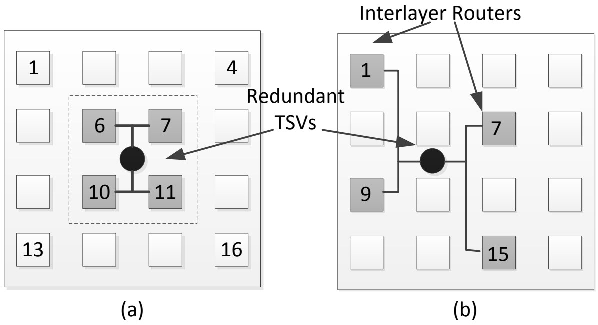

Figure 4.4. Router placements on the core layer for a 16-core system. (a). The interlayer routers are placed in the middle; (b). The interlayer routers are distributed evenly across the core layout.

4.2.3.1 Router Placement

As shown in Section 4.1, our target system is the core-to-cache/memory stacking using TSVs for core to memory communication. There are several possible router placements strategies to support vertical connection. Based on our synthesized result, the area of a 3D-capable router is almost twice the area of a 2D router. The synthesis is done with 45*nm* library and the related router areas are shown in Table 4.1. Therefore, it will incur significant area overhead to have each router 3D connected. We assume the ratio between the number of 3D routers and 2D routers as 1:3. The possible router placements on the core layer are depicted in Figure 4.4 for a 16-core system. The other layers have the same router placement as the core layer. In order to reduce the wire routing complexity, redundant TSVs are located in the same region and connected to routers with switches.

In Figure 4.4(a), four interlayer routers are placed in the middle, which results in the shortest latency according to a previous study [126] and provides balanced routing distance to redundant TSVs. However, this design stresses these four routers not only from their own vertical communications, but also from the horizontal traversals ejected from neighboring 3D routers. For example, if a flit travels from a layer to this layer through router 10, and its destination is router 16, then through XY routing algorithm, the horizontal link between router 10 and 11 is requested before the flit is forwarded to other 2D routers. Compared to the centralized placement, the placement in Figure 4.4(b) has less horizontal stresses because each 3D router is directly connected to at least two 2D routers. Thus, the flit can be easily forwarded to the 2D routers after the vertical traversal. In our design, we mainly focus on the second placement.

4.2.3.2 Routing Algorithm and Timing Analysis

Lots of previous studies have explored the 3D routing methods [134, 135]. In this work, we use the straightforward dimension order routing algorithm [135]. Whenever the source and destination are within the same layer, normal XY routing is used to avoid deadlock. If interlayer communication occurs, the router will find its nearest interlayer router to perform vertical traversal. Although the exploration of routing algorithm is beyond the scope of this work, other routing algorithms, such as bandwidth-aware routing algorithms [132], can also be applied in our design.

The virtual channel router is a five-stage pipeline design and the clock frequency is determined by the slowest stage. We synthesize the components of each stage and perform timing analysis. The longest pipeline stage is the virtual channel allocation, which takes $1.77ns$. In comparison, our AL allocator requires $0.73ns$ to perform switch allocation. Therefore, adding the AL arbitrator has no influence on the router frequency as it can be executed in parallel with any pipeline stage.

4.2.3.3 Flow Control

Due to the buffer size limitation at the input side, flow control methods send the buffer availability of the downstream router to the upstream router to avoid packet loss. In our work, we use the prevalent credit-based flow control mechanism [133].

In the credit-based flow control, the upstream router keeps a counter of free buffers for each downstream virtual channel. Whenever the upstream router forwards a flit, the corresponding counter decreases by 1. The downstream buffer is full if the counter reaches zero, then the upstream router waits until there is a free buffer. Every time the router forwards a flit to its downstream router, it sends a credit to its connected upstream router, incrementing the upstream buffer counter. In this way, the downstream routers can inform its upstream router about the buffer availability. Even though in our design, it is possible that the upstream router forwards two flits per cycle, the target virtual channels are different. Therefore, the flow control strategy can keep unchanged.

Table 4.1. The synthesized area and power of 2D/3D routers, crossbar, and AL arbitrator.

Component	**Area** μm^2	**Dynamic Power** mW	**Static Power** μW
2D Router	46422.85	16.044	1040.1
3D Router	84669.13	26.85	1924.8
5×5 Xbar	641.86	0.27	13.97
7×7 Xbar	1324.68	0.63	29.99
9×7 Xbar	1688.57	0.81	38.16
AL Arbitrator	122.89	0.065	2.89

In our design, even though the router can send two flits to one output, the flow control mechanism can keep unchanged. It is because we restrict one input to forward only one flit, and the upstream router for each input is different. As a result, each counter in a router can at most increase or decrease by 1 per flit cycle.

4.2.3.4 Availability of Spare TSVs

In our work, the design relies on spare TSVs to provide additional vertical links. The number of spare TSVs is determined by the TSV failure rate, fault mode, and redundancy ratio. It is possible that the spare TSVs count is smaller than a flit size. In this case, we simply disable the AL arbitrator and run the router in its original mode. On the contrary, if the redundancy ratio is high that the number of spare TSVs is multiple times of a flit size, there is opportunity to transmit more than two flits simultaneously over the vertical links, which further improves the network throughput. Even though this case is not evaluated in this work, our design can be easily extended to account for more spare TSVs.

4.3 Simulation Result Analysis

In this section, we first present our experiment configurations. Then the area and power comparison of the original and proposed design are listed. For the performance evaluation, we modify the uniform synthetic workload to mimic core-to-cache/memory communications and report the average packet latency. At last, we explore how the availability of spare TSVs influence the network performance.

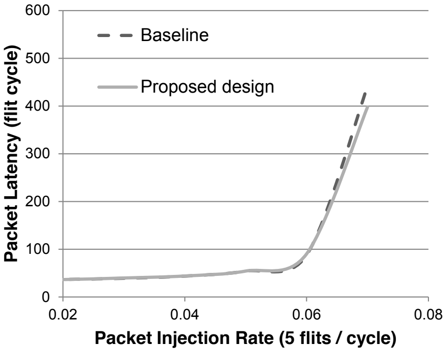

Figure 4.5. Packet latency of uniform traffic in a 4×4 mesh network.

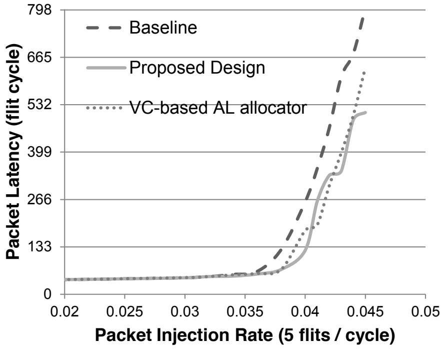

Figure 4.6. Packet latency of interlayer traffic in a 4×4 mesh network.

We use the cycle-accurate NoC simulator Booksim [136] for performance simulation and implement our designed components in the simulator. The performance of two mesh networks, 4×4 and 8×8 nodes, are evaluated. The number of layers in the 3D stacking is two by default. Since each interlayer router is responsible for vertical communications of other four surrounding 2D routers, the virtual channel

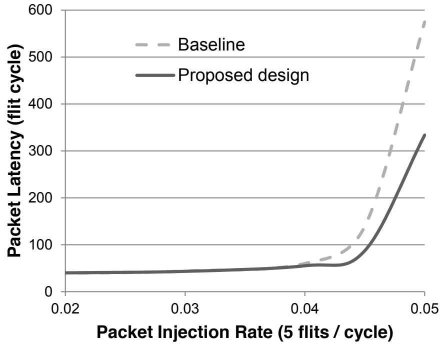

Figure 4.7. Packet latency of weighted interlayer traffic in a 4×4 mesh network.

is a highly competitive resource. Based on our simulation, when each port contains 4 virtual channels and 4 buffers per VC, the router will be congested at very low injection rate (less than 0.04 packet/cycle) due to the bottleneck of VC allocation. Nevertheless, our synthesized results show that doubling the buffer size makes the router area increase by almost 2.7 times. To balance the performance and area, we configure the router to contain 8 virtual channels per port with 2 buffers per VC. In our design, we assume that each packet size is 64B and the flit size is 128 bits. Adding the header flit, one packet is separated into 5 flits.

4.3.1 Area Evaluation

We synthesize the RTL description of the virtual-channel router [136] using Design Compiler [137] and 45nm NanGate cell library [118]. The area and power are listed in Table 4.1.

The interlayer router consumes almost twice the area of 2D router while the total buffer ratio is 7:5. The area overhead comes from buffers and other control logics. The crossbar only occupies 1.4% and 1.6% of total router area in 2D and 3D cases, respectively. In our design, the crossbar contains two additional output links, therefore, the area increases about 27.5% compared to original crossbar. In terms of the whole interlayer router, the area overhead is smaller than 0.4%. The power consumption is roughly proportional to the area, thus, the power overhead of the modified crossbar to the router is about 0.67%. The area of AL arbitrator is about 0.15% of a single router. Since the AL arbitrator is shared by four interlayer routers, the area overhead is negligible.

4.3.2 Performance Evaluation

Three types of synthetic workloads are generated to mimic the network traffic in core-to-cache/memory systems: *uniform traffic*, *interlayer traffic*, and *weighted interlayer traffic*. The source and destination are randomly selected and the injection rate of each node are almost the same in the uniform traffic. The interlayer traffic models the case that all traffics are interlayer, and this kind of traffic pattern maximizes the stress of interlayer routers. The last one is the weighted interlayer traffic, which generates the interlayer traffic according to a customized ratio. Among these three traffic patterns, the last one is the closest to real cases since coherence messages are transmitted within the layer while cache/memory requests are traveling through vertical links.

The ratio of interlayer communication is set to 0.7 in the weighted interlayer traffic. In the interlayer traffic, we show the performance with the virtual buffer occupancy based AL arbitrator. The packet latency versus injection rates are shown. When the interlayer communication percentage is low, as shown in the uniform traffic (Figure 4.5), the network saturates at higher injection rate. The interlayer traffic (Figure 4.6) has the lowest saturated injection rate due to the highly congested interlayer routers. The proposed design can reduce the packet latency when the injection rate is high in all three traffic patterns. The performance improvement is more obvious in high interlayer communication patterns, such as the interlayer traffic and weighted interlayer traffic. For example, in the weighted interlayer traffic, when the injection rate is 0.05 packet per cycle, the latency in baseline is above 500 cycles while it is only around 300 cycles in the proposed design. Even though considering the virtual channel buffer occupancy can reduce the input congestion of downstream router, it reduces the chance of utilizing additional links. Therefore, the latency is larger than the design with basic AL arbitrator as shown in Figure 4.6. Usually, the high vertical request count implies that the downstream router is also stressful. Comparing filling two routers with waiting flits and filling only the downstream router, the former one may lead to congestions on all the connected routers in both layers, while the latter only results in congestions in one layer.

To further analyze how additional links help reducing the network delay, we examine the average flit queuing time of routers in the bottom layer for the weighted interlayer traffic. The injection rate is set to 0.035 packet/flit cycle, before the network delay becomes saturated. In the proposed design, the average queuing latency of 2D routers are reduced, whereas the interlayer router queuing delay increases. It is because interlayer routers receive more flits from the corresponding upper router, increasing the virtual channel allocation and switch allocation delay.

The corresponding input buffer read/write ratios of one interlayer router (Router 7) and one 2D router (Router 11) are shown in Figure 4.8 and Figure 4.9, respectively. The input marked *self* represents the input port from the attached node

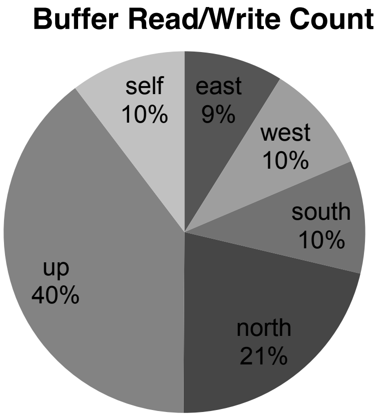

Figure 4.8. The input buffers read/write ratio analysis of the interlayer router.

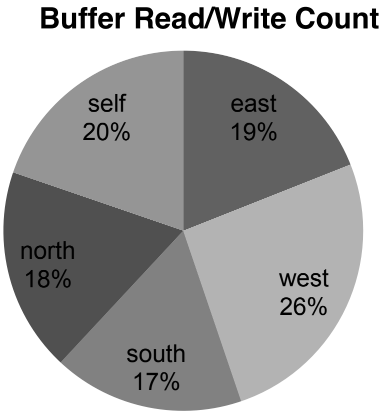

Figure 4.9. The input buffers read/write ratio in different directions of the 2D router.

to the router. Since router 7 is on the bottom layer, only the input buffer storing messages from upper layers has been used. The workloads in each directions are almost balanced in the 2D router. In contrast, the interlayer router shows significantly large demands on the vertical direction. The traffic from upper layers occupies almost 50% of the total traffic in that router.

The corresponding average network dynamic power consumption is shown in Figure 4.10. As shown in the figure, when the injection rate is low and the performance improvement is not obvious, the proposed design has similar power consumption as the baseline. With higher injection rate, the proposed design has higher power consumption due to the accelerated flit transmission. Since our proposed design can reduce the workload execution time, the final network energy can be reduced by 6.51% at 0.05 injection rate.

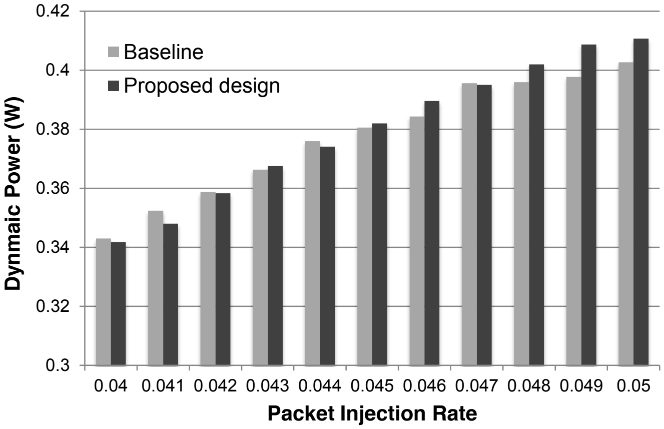

Figure 4.10. The dynamic power consumption comparison between baseline and proposed design.

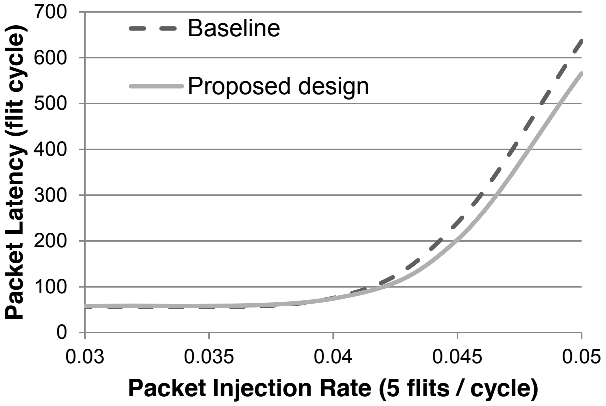

Figure 4.11. The average packet latency in 8×8 mesh network with the weighted interlayer traffic.

The average packet latency with 8×8 mesh network is shown in Figure 4.11 with the weighted interlayer traffic. In this network, there are total 16 interlayer routers and 4 groups of spare TSVs. The performance improvement in this network is not as distinct as in 4×4 mesh network due to the decreased network congestion.

4.3.3 Layer Sensitivity

Different number of layers in the 3D stacking result in various degree of congestion. In this set of experiment, the network performance is examined with 2, 3, and 4 layer stackings. The topology is 4×4 mesh and the traffic is the weighted interlayer pattern with injection rate equals to 0.35. Table 4.2 shows the results

Table 4.2. Average packet latency comparison of different layers in 3D systems.

Layer	Baseline (cycle)	Proposed (cycle)
2	48.159	49.026
3	60.925	54.99
4	795.68	401.86

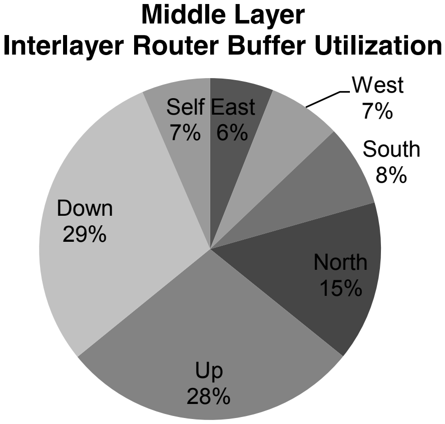

Figure 4.12. The input buffers read/write ratio of the interlayer router in the middle layer.

of simulation. As expected, with increased number of layers, the vertical channel is more congested, making our proposed design more favorable.

Similarly, we examine the input buffer read/write count of one interlayer router in the middle layer with the 3 layer stacking. It can be seen from the result that more than 50% of total traffic are interlayer traffic with 28% of traffic to the lower layer and 29% to the upper layer. Therefore, for interlayer routers, especially for routers in the middle layer that are relaying message from both upper and lower layers, increasing the vertical bandwidth can accelerate the transmission.

4.3.4 Failure Mode Sensitivity

The proposed design relies on the availability of spare TSVs. If the TSV failure rate is high, the spare TSVs cannot be used as additional channels. In most cases, the TSV fails due to the manufacturing limitations and these failures are irreversible and can be determined once the fabrication is finished. Therefore, we study the network latency when one group of spare TSVs is not available. The topology

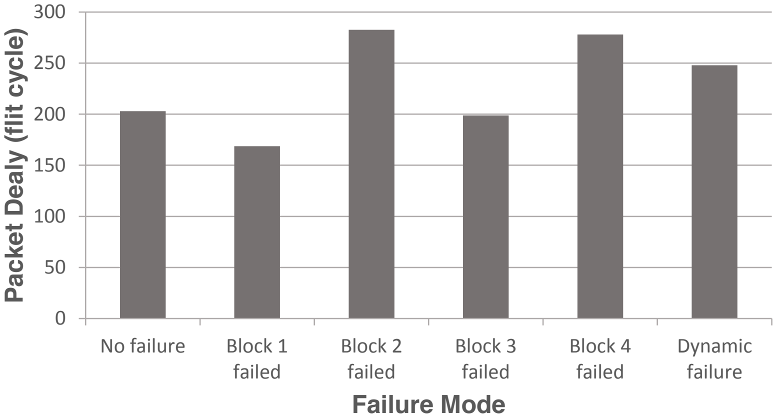

Figure 4.13. The average packet latency with different locations of unavailable spare TSVs and dynamic failure modeling.

is 8×8 mesh network since there are four groups of interlayer routers (every 4×4 router block contains one group). We define these four groups of routers according to their location as Block 1 to Block 4. Therefore, in the TSV failure sensitivity study, we model five cases: no TSV failure (all spare TSVs can be used as additional links), block 1, block 2, block 3, or block 4 has failure (one group of spare TSVs cannot be used). Figure 4.13 shows the average packet latency of these five cases.

Because of the router placement and routing algorithm, the loads among blocks are not balanced. From our observation, we find that the load in the block 1 and 3 is the lighter than the block 2 and 4. Therefore, when the spare TSVs in block 2 and block 4 cannot be used as additional links, the performance degradation is severe. However, when the failure is in block 1, the average packet latency is reduced compared to the no failure case. Because the congestion is not heavy in block 1, whereas our proposed design leads to virtual channel allocation or switch allocation delay overhead in the downstream router, which offsets the delay reduction from accelerating flit transmission.

In addition to the manufacturing limitation, the TSV may fail due to thermal stresses and electromigration. These failures occur during system operation and are closely related to the TSV utilization. We capture the TSV dynamic failure by monitoring the transaction frequency of each vertical link and assign each TSV array with a failure probability according to their utilization. This process continues until there is no spare TSVs. The average latency is shown as the last bar in Figure 4.13. At the beginning of operation, all the spare TSVs are capable for additional links, therefore, the final package latency is better than the case of block 2 and block 4 failures. Then due to the high utilization, spare TSVs in block 4 become unavailable before others. The packet latency is then increased due to the vertical request congestion.

4.4 Summary

Due to the increased communication complexity and variation in 3D many-core systems, NoCs with design-time defined channel width may not be able to fulfill the burst interconnect requirement. In 3D designs, redundant TSVs are the most common solution towards low yield. However, these allocated redundant TSVs may not be used for repair if the original TSV is functional. In this Chapter, we propose to use these allocated but not used redundant TSVs as additional vertical links to relieve the network congestion from the burst interlayer traffic. The simulation results show that our proposed design can reduce average packet latency with negligible area and power overhead when the network congestion is high.

Chapter 5

Electromigration Lifetime Analysis in TSV Arrays

In previous chapters, the cost-aware design methodologies are introduced. Besides the cost, the reliability problem in 3D ICs attracts many researchers' and industrial attention, as the reliability is closely related to the success of high volume production. In the following chapters, the reliability-aware design methodologies are illustrated covering three different aspects: interconnect reliability due to electromigration, signal integrity induced by crosstalk, and chip-level mechanical reliability due to thermal stresses.

Electromigration (EM), which refers to the migration of metal atoms in response to an electric field in a conductor, has been proven to be one of the major factors of interconnect failure [138, 139][1]. The interconnect failure can affect the chip performance or even cause malfunction. As a result, EM lifetime analysis and corresponding alleviation techniques are critical to guarantee the chip lifetime and satisfy design specifications. The factors that influence EM lifetime of interconnects have been unveiled in previous studies [140, 141], emphasizing the impact of current density on conductor's EM mean-time-to-failure (MTTF).

Although 3D ICs provide various benefits over traditional 2D designs, the design of power delivery network becomes more challenging than 2D counterparts due to the aggressively increased power density and limited power pin count in the packaging. Therefore, considering the higher current density requirement on power/ground (P/G) TSVs for the power delivery, it can induce severe EM problems on P/G TSVs [142]. Moreover, as TSV manufacturing technology evolves, smaller TSVs are expected to reduce RC delay and footprint, which may result in even higher current density. Therefore, the reliability problem caused by EM degradation is exacerbated in the power supply network of a 3D chip.

---

[1] This work is published as "TSV power supply array electromigration lifetime analysis in 3D ICs" on GLSVLSI 2014.

In order to guarantee the signal integrity, a TSV array that consists of multiple TSVs is commonly used for single signal delivery [143]. Similarly, the TSV array is applied for the power delivery due to the limited current delivering capability of a single TSV. Unfortunately, prior studies on EM lifetime modeling and analysis mainly focus on individual TSVs, leaving the EM lifetime analysis on such TSV arrays unexplored. Different from previous work, we model P/G TSV arrays and perform the EM lifetime analysis on these TSV arrays.

The EM lifetime analysis of TSV arrays becomes complicated because of two unique challenges. The first challenge is the **Current Distribution** among TSVs within the same array. Current density is an important factor to determine the lifetime of an individual TSV and different current distributions can result in different current densities in an array. Therefore, it is necessary to know how the current is distributed within a TSV array. The second issue is more important, which is related to the **Current Correlation** among TSVs. Obviously, there is a tight correlation between the current density of each TSV within an array. For example, the failure of one TSV can cause current redistribution among remaining TSVs in the same array. Consequently, the remaining TSVs would carry a larger current that in turn shortens their EM lifetimes. In this work, we propose an analysis framework to estimate the EM lifetime of TSV arrays and study the impacts of design parameters on the EM lifetime of TSV arrays.

5.1 Background and Motivation

5.1.1 Motivation

In 3D designs, the power between two stacking tiers is transmitted through TSVs. The reliability of P/G TSVs can influence the power integrity, and eventually affect the circuit functionality. In general, there are three major reasons to explain why P/G TSVs suffer from severe EM problems. First, 3D ICs have smaller footprint than 2D counterparts, which means higher power density and less pin-outs. Therefore, the current density per pin is increased. Second, in 2D P/G networks, wire sizing technique is used to alleviate the EM degradation. Due to the large area overhead, this technique cannot be applied to TSVs. Furthermore, to reduce the parasitics and area overhead, the TSV size tends to decrease with technology scaling, resulting in even higher current density. Third, in addition to supply power to the tier they are bonded to, TSVs are also responsible for power supply to upper tiers that are far from power pins. As a consequence, TSVs that are closest to power pin experience the largest current and they are more vulnerable to EM problem.

Figure 5.1 shows a simple example of a 3D stacked architecture with P/G TSV arrays. Multiple dies are stacked face-to-back and the voltage sources are placed in the package level on the bottom. One P/G TSV from the array is zoomed in and the current flows are shown in the figure. I_{TSV} represents the current going through the TSV and I_1 to I_4 mean the current on connected 2D metal wires. I_{up} denotes the total current supply to upper tiers. Based on Kirchhoff's current law, $I_{TSV} = \sum_{i=1}^{4} I_i + I_{up}$. From the equation, we find that I_{TSV} is at least four times larger than the average current in metal wires.

In addition to the current magnitude, the conductor cross section area is another factor in current density calculation. In recent studies, the reported width of metal wires is $2\mu m$ and the thickness is $1\mu m$, while the TSV diameter is about $5\mu m$ [144, 145]. The area ratio between TSV and metal wire is about 10:1, which means the current density ratio in TSV and metal wires is about 1:2.5. Nevertheless, from the DC current crowding study [144], researchers find that current density on the edge of TSV is nearly twice than that inside the TSV. Combining the current and area analysis, we conclude that the current density on TSV's weakest points and metal wires are almost the same. Moreover, high aspect ratio TSV is preferred for future IC designs and TSV's area is continuously reduced, making the current density on TSV even larger than on metal wire. From the analysis above, we find that the EM analysis in TSVs is not trivial.

Due to the resistance on the metal wires, individual TSVs in the array suffer

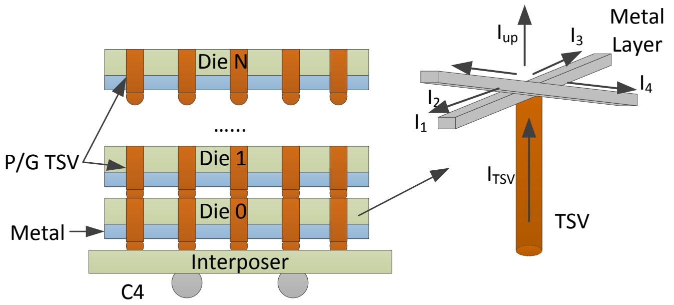

Figure 5.1. The overview of stacked dies with 3D P/G network.

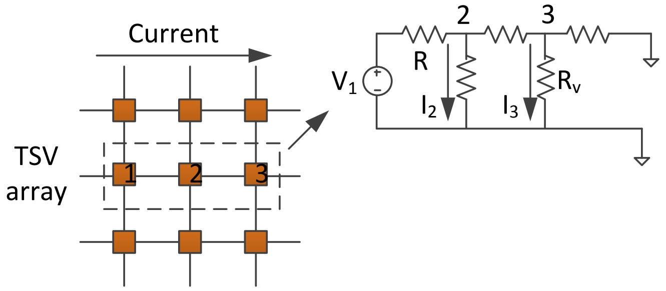

Figure 5.2. Top view of TSV array with unevenly distributed current.

from different current loads. As shown in Figure 5.2, we assume TSV1 is connected to power pin with supply power V_1. The resistances on metal wires and TSVs are R and R_v, respectively. In this example, we only consider one current flow direction. Since the system is symmetric, superposition can be applied for whole system analysis [146]. Therefore, I_2 (current goes through TSV2) is $\frac{V_2}{R_v}$, while I_3 (current goes through TSV3) equals to $\frac{V_2}{R_v} \cdot \frac{R_v}{R+R_v}$, which is smaller than I_2. Since some TSVs carry more current than others, the EM lifetime of these TSVs are shorter. After these TSVs' failure, the current is redistributed and the rest TSVs need to carry more current which will accelerate their EM degradation. Based on the current distribution and current correlation, we should re-calculate the TSV array EM lifetime since the simple assumption of even current distribution leads to erroneous prediction of EM lifetime.

5.1.2 Related Work

The EM issues in single TSV have been studied with various configurations. By varying the stress conditions and thickness of metal wires that are connected to the TSV, the EM behavior has been analyzed. The results show that voids are formed in TSV, especially at the interface between the TSV and metal wires, resulting in increased TSV resistance [147]. The similar conclusion is drawn that the resistance shifting is easily found at the material interface [148]. With respect to the resistance increment due to EM on the interface, an analytical model is proposed [149] and it shows that a single TSV's MTTF is in accordance with Black's equation [140] which has been widely applied to analyze the EM lifetime for metal wires.

The EM lifetime analysis focusing on TSV arrays in a P/G network has not been explored yet in previous studies. In the 2D realm, multi-vias structure are used to support tremendous current density. The analysis of how multiple vias influence the current distribution and EM performance has been preliminarily explored. The results from the experiment show that the current distribution within the array is uneven. Moreover, when the current stress remains the same and more vias are in the array, the EM lifetime is longer with smaller current distribution variation [150]. Even though the average current density in the uneven distribution is smaller, the via array under uneven current distribution has worse MTTF than the same structure with even current distribution [146]. Intuitively, the similar problem could occur in TSV arrays.

To our best knowledge, this is the first work to analyze the EM lifetime of TSV arrays considering uneven current distribution. In this work, we present a comprehensive current distribution and EM lifetime analysis flow on TSV arrays. We also find out effective approaches to extend the EM lifetime of TSV arrays by exploring a better trade-off between EM lifetime and design cost.

5.2 TSV Array Current Density and EM Lifetime Simulation Mechanism

In this section, the 3D power network model is introduced first. This model demonstrates connections between TSVs, 2D power grid, and voltage supply pins. Based on the model, the corresponding voltages of power grid nodes and TSVs are calculated with modified node-based fast algorithm [151]. Then the framework to calculate the current density and the EM lifetime of TSV arrays is explained. In general, the EM analysis framework contains two stages as shown in Figure 5.3. The first stage concentrates on the current distribution analysis. The calculated current densities are then used as inputs for the second stage to estimate EM lifetime of the TSV array.

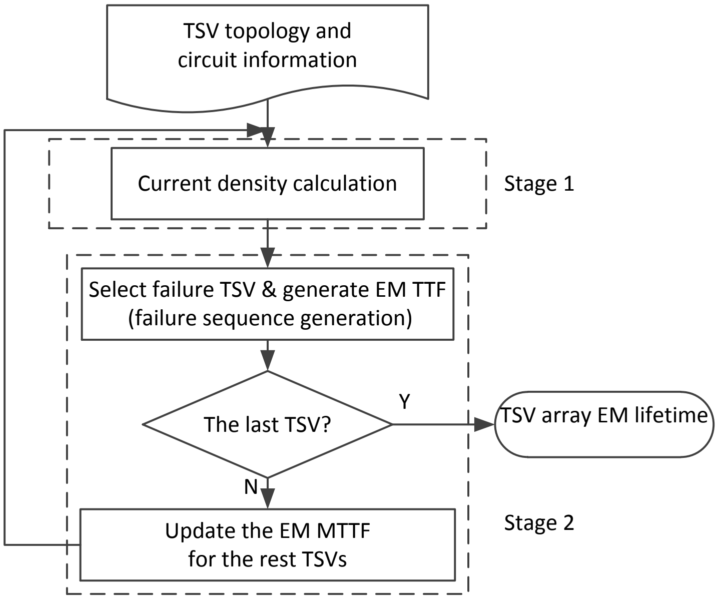
TSV topology and circuit information
Current density calculation
Stage 1
Select failure TSV & generate EM TTF (failure sequence generation)
The last TSV?
Y
TSV array EM lifetime
N
Update the EM MTTF for the rest TSVs
Stage 2

Figure 5.3. The EM lifetime analysis framework of TSV arrays consists of current distribution calculation and array lifetime estimation.

5.2.1 Power Grid Model

As elaborated in the previous section, P/G TSVs are suffering from severe EM degradation. Furthermore, the current flow direction is usually constant during circuit operations, resulting in continuous driving forces for atom migration. Previous studies have found that the healing effect from bi-directional current flow can extend the EM lifetime of power supply networks [152]. This phenomenon can be found in power gated circuitries or multi-storied power supply networks in 3D designs [153, 154]. Nonetheless, the bidirectional current flow is rarely observed in our designs. As a consequence, we mainly focus on the unidirectional current stress in our power network model.

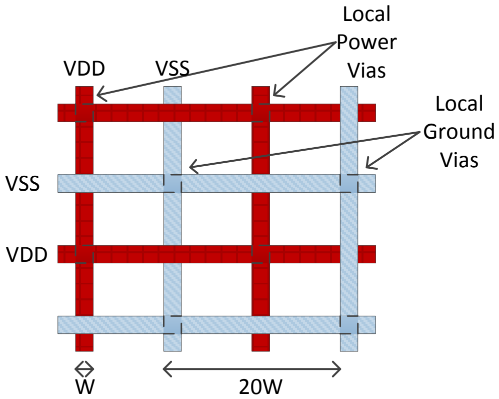
Local
Power
Vias
VDD
VSS
Local
Ground
Vias
VSS
VDD
W
20W

Figure 5.4. The 2D planar P/G grid network model on each tier.

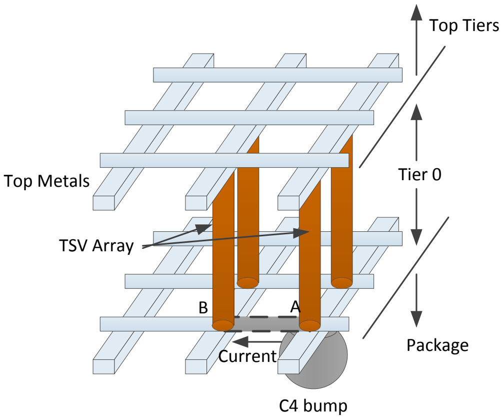
Top Tiers
Tier 0
Top Metals
TSV Array
B
A
Current
Package
C4 bump

Figure 5.5. The portion of 3D power supply network with one TSV array (only VDD is shown).

The 3D power network model contains two parts: 2D planar power grids and vertical power connections. We assume the power grid structure on each tier as shown in Figure 5.4. TSVs connect the global power grid between two adjacent tiers. The two orthogonal metal layers are used for power or ground rails. Power rails (in dark color) and ground rails (in light color) are placed alternately and local vias connect the corresponding rails on two metal layers. Assume the global power grid occupies roughly 10% of routing resources [155] and the width for top metal after EM-aware sizing is W, therefore, the distance between two power rails is 20W (calculated from $\frac{W}{10\%/2}$).

The vertical power connections are composed of C4 bumps, micro bumps, and TSV arrays. Since power and ground networks are symmetric, throughout this work, we mainly focus on power network analysis. The 3D power supply network

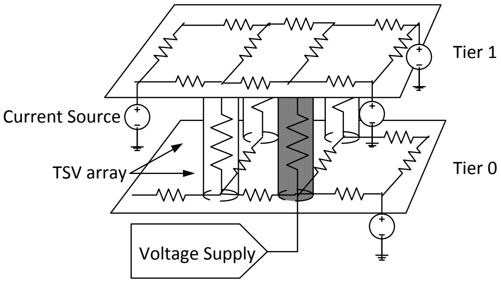

Figure 5.6. The corresponding resistance path of 3D power supply network.

with one off-chip power pin is shown in Figure 5.5 as an example. Note that usually multiple off-chip power pins are used for full chip power supply and each power pin connects to one TSV array. C4 bumps can represent off-chip P/G pins and micro bumps work as intermediate connections between TSVs and metal wires. C4 bumps and micro bumps are considered as ideal conductors in our work. In the design, only one TSV (*center TSV*) in the array is directly connected to one C4 bump while other TSVs (*peripheral TSVs*) are connected to the center TSV with metal wires as proposed by previously study [156]. The kind of power delivery network design is better to help alleviate IR drop between tiers. Therefore, we follow the similar design in which only the center TSV is connected to the voltage supply while all peripheral TSVs are susceptible to voltage fluctuations on metal wires.

Constant unidirectional current stress is applied, therefore, the inductance and capacitance are irrelevant to our model, leaving resistance the only parasitic parameter that should be concerned. Due to the resistance network, current is unevenly distributed among TSV arrays. For example, the resistance along the metal wires causes voltage drop from point A to point B in Figure 5.5. In our framework, the value of voltage drop between two neighboring TSVs is extracted from HSPICE simulation and remains constant during the following EM analysis. The example of extracted resistance network is shown in Figure 5.6. Four TSVs are employed to form a power supply array and only the TSV at lower right corner is directly connected to the power pin as the center TSV. Note that the upper left TSV has the lowest voltage due to IR drop. TSVs and metal wires are modeled as resistors and the devices powered by the TSV array are abstracted as current sources which are predetermined or extracted from real designs.

5.2.2 Current Density Calculation

Due to the dependency of TSV's EM lifetime on the current density, the current analysis is performed to generate the current distribution map. In recent studies, in additional to current density, other EM driving forces including atomic concentration gradients, thermal gradients, and stress gradients are revealed [141]. The detailed EM lifetime calculation considering these factors requires significant computation overhead and it is infeasible to model gradual failure in TSV array. In this work, we focus on the impact of current distribution on the EM lifetime, therefore, we only use Black's equation to estimate the MTTF.

The current density on TSV is calculated from the voltage difference. Therefore, the voltage analysis is performed first when the resistance network is known. The voltage analysis contains two steps in the proposed framework: 2D power grid voltage analysis and 3D power propagation. We modify the node-based fast algorithm [151] for 2D power grid voltage analysis and voltage propagation method [157] for 3D TSV voltage analysis.

Different from [157], we place the off-chip power supply on the bottom tier and use distributed TSV array topology. As a result, given that the center TSV is directly connected to the supply voltage, the voltage calculation starts from the top tier and continues until the voltage propagates to the bottom tier. The initial values contain reference voltages of the peripheral TSVs on the bottom. The reference voltages are calculated based on the normal distribution with predefined variation (represented as *scale factor*) extracted from HSPICE simulations. The algorithm begins with the node-based fast algorithm [151] for the voltage analysis of power grid nodes on each tier. Once the voltage on every grid node is available, the current that goes through a TSV can be calculated by applying Kirchhoff's current law $\sum_{k\in N_i} G_{ki}(V_k - V_i) = I_i$, where i, k refer to node IDs, and N_i is the neighboring nodes of TSV$_i$. G_{ki} is the conductance between node k and TSV$_i$. V_i stands for the voltage on TSV$_i$, and I_i is the calculated current flowing through TSV$_i$.

The corresponding voltage calculation process is elaborated in Algorithm **??**. Note that all voltage and current symbols in this algorithm are vectors that store the corresponding values for each TSV in a TSV array. The initialization voltage value on the top tier (V_N) equals the the reference voltage vector (V_{ref}) that is

Algorithm 2 The algorithm outline for TSV array current density calculation using voltage propagation.

```
1: set $\varepsilon \ll 1$;
2: $V_{ref} = V_{DD}$ * Scale Factor;
3: $V_N = V_{ref}$;
4: while MAX($V_{diff}$) $> \varepsilon$ do
5:     for Tier $t = N$ to 1 do
6:         Row based voltage analysis with $V_t$ [151];
7:         Calculate $I_t^{tsv}$ with Kirchhoff's current law;
8:         $V_{t-1} = V_t + \sum_{k=t}^{N} I_k^{tsv} * R_{tsv}$;
9:     end for
10:    $V_{diff} = V_{ref} - V_0$ ;
11:    $V_N = V_N - V_{diff}/N$ ;
12: end while
13: return TSV current vector $< I_N^{tsv}, I_{N-1}^{tsv}, ..I_1^{tsv} >$;
```

calculated in line 2. After initialization, the algorithm enters the iterations for the voltage and current calculation in a tier-by-tier fashion. In each iteration, the voltage obtained from the previous iteration is used as the input. The current in a TSV array are calculated by applying row based voltage analysis [151] and Kirchhoff's current law (line 6-7). Afterwards, the TSV voltage of next tier is calculated with cumulative TSV IR drops from the previous tiers (line 8). Once the voltage propagation completes, there may have voltage differences (V_{diff}) between the bottom tier TSV voltage (V_0) and reference voltage (line 10). Consequently, the voltage on the top tier should be tuned accordingly (line 11). As the voltage of the top tier changes, the voltages on all tiers should be re-evaluated iteratively. The whole algorithm terminates with a convergence condition when the maximum voltage difference is beyond a predefined threshold (ε). Tuning the top tier TSV voltage with V_{diff} divided by the tier number (N) can accelerate the algorithm convergence. Finally, TSV currents in each tier are returned for the EM lifetime analysis in next stage.

5.2.3 Array EM Lifetime Calculation

In order to explore the longest possible lifetime, we assume that **a TSV array fails if and only if the last via in the array is worn out due to EM**

effect. Due to the current distribution and current correlation between TSVs in the array, the failure sequence is of great importance to determine the array EM lifetime. However, the searching space of the failure sequence is too large to be fully explored. Previous studies have shown that TSV's EM lifetime follows a lognormal distribution with the MTTF calculated from Black's equation [139,147]. Therefore, we apply the lognormal distribution for the calculation of a single TSV's EM lifetime and combine it with Monte Carlo approximation to generate the EM lifetime distribution of a TSV array.

The EM effect is memorizable, which means that the past behavior has impact on the future performance. Therefore, the stress time translation is necessary to take the stressed history into account after current redistribution. The following example can explain the process of stress time translation. If one TSV has operated under current density of $20m\text{A}/\mu m^2$ for 2s, the EM stressed time can be translated into 1.28s with current density of $25m\text{A}/\mu m^2$ after current redistribution. We leverage the translation rule $\left(\frac{I_{m-1}}{I_m}\right)^2 = \frac{t_m}{t_{m-1}}$ as proposed in previous work [146], where I and t denote the current density and stress time; m represents the timing sequence.

After current densities are generated for the TSV array through the voltage propagation process, one of possible failure sequences can be determined in the following procedure. First, we select one TSV in each iteration and make it fail with weighted probabilities determined by their current densities in the program. The larger current density one TSV has, the higher probability that it will be selected. The MTTF of the selected TSV is then calculated and the EM failure time is obtained following the lognormal distribution. Then, the stressed time for the remaining TSVs are translated according to the translation rule. The current density calculation process is applied for the rest TSVs to establish the new current density map. The iteration continues until the TSV array is failed due to EM degradation. To this end, one sample of array EM lifetime is successfully calculated. Monte Carlo estimation with massive samples is used to generate the MTTF and lifetime distribution for the TSV array.

5.3 TSV Array EM Lifetime Analysis

This section shows the TSV array EM lifetime analysis results with various design parameters. We first analyze how the current density distribution affects the estimation of EM lifetime, comparing to cases when the current is assumed to be evenly distributed. Furthermore, the TSV filling material can determine both the TSV resistance and activation energy in Black's equation, resulting in different EM lifetimes. Consequently, we then examine the EM lifetime distribution when we use different TSV filling materials. We also study the impact of TSV diameters, TSV counts, and current loads on the EM behavior. In terms of area overhead, we further consider the TSV count/size tradeoff if total array area is fixed.

For each experiment, we capture massive Monte Carlo samples to calculate the EM MTTF and generate the EM lifetime distribution. From our experiments, we find that accurate results can be obtained when the number of samples is larger than 1000. Therefore in the following analysis, considering the trade-off of estimation accuracy and computational overhead, 4000 Monte Carlo samples are generated for each experiment. A two tiers stacking chip is used as the design target. The TSV diameter is $5\mu m$ and the height is $30\mu m$ [144]. Since accurate dynamic current is not available at the early design stage, the current load of each node in the power grid is assigned assuming an universal activity factor. The total current load remains constant for all experiments for fair comparison. Experiment results are normalized to the case with one TSV under the same current load.

5.3.1 TSV Array EM Lifetime

For a single TSV, the EM lifetime follows a lognormal distribution. In our experiment, we apply this distribution for individual TSVs in an array for EM lifetime estimation of TSV arrays as well. In this experiment, the target P/G TSV array contains 4×4 TSVs. Figure 5.7 shows the distribution of TSV array EM lifetime with the proposed analysis framework. From the figure, we can observe that the TSV array lifetime also follows the lognormal distribution. We also conduct experiments on different TSV counts with the same current load. The comparison result is shown in Figure 5.8. With more TSVs, the EM lifetime is extended and the distribution variation is larger since the number of possible failure sequences

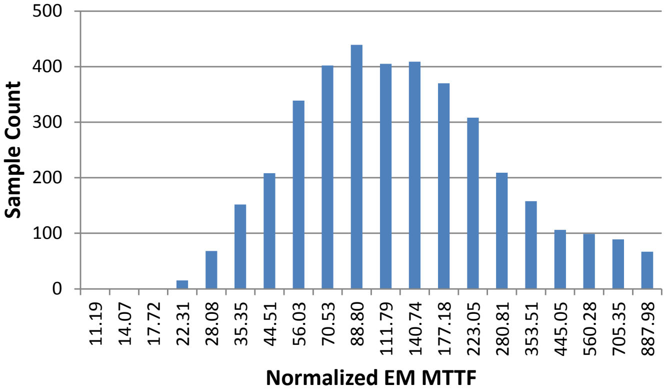

Figure 5.7. 4×4 TSV array lifetime distribution.

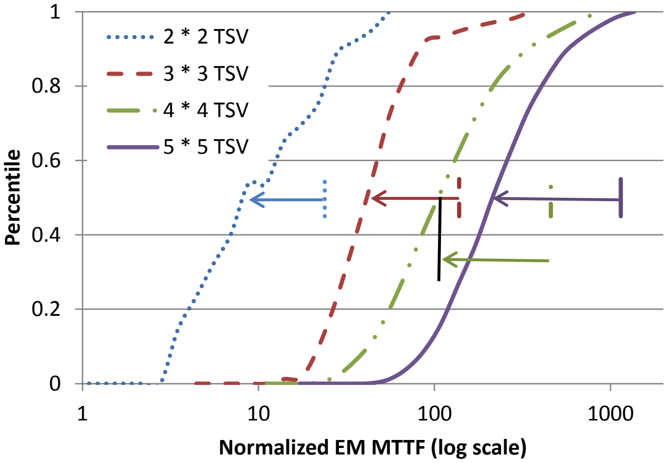

Figure 5.8. EM lifetime distribution with different TSV arrays. The y-axis shows the number of samples/percentile, and x-axis shows the normalized EM lifetime in log scale.

is increased, making it harder to predict the exact EM failure time.

When an even current distribution is assumed on the TSV array, the EM lifetime would be overestimated. The results show that for four TSV array configurations (2×2, 3×3, 4×4, and 5×5), the EM MTTF in uneven current distribution is only 36.79%, 32.02%, 24.38%, and 18.89% of that in even current distribution as shown along the arrow in figure, respectively. The percentage goes down with larger TSV array because the current distribution is more uneven.

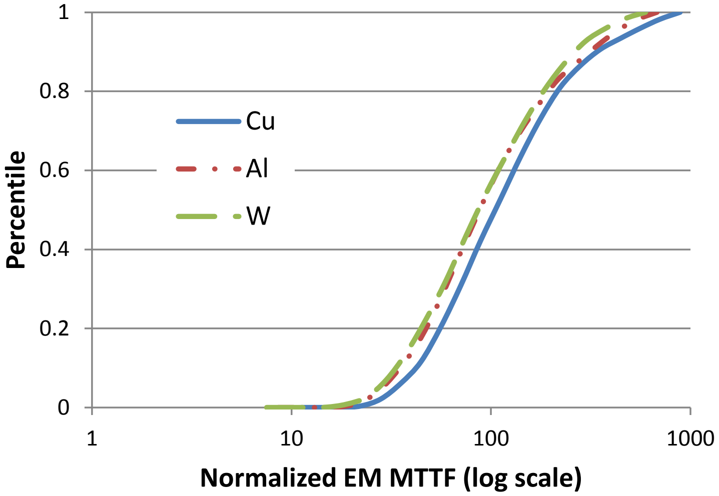

Figure 5.9. TSV array EM lifetime distribution with different filling materials: Cu, Al, and W. The y-axis shows the percentile of the TTF distribution, and x-axis shows the normalized lifetime in log scale.

5.3.2 TSV Filling Material

The common filling materials for TSV include copper (Cu), aluminum (Al), and tungsten (W) [158] with varying resistivity and activation energy. In this section, we study the impact of TSV filling materials on the EM lifetime. Among these three materials, copper has the smallest resistivity. Due to different process steps, tungsten TSVs normally have higher aspect ratio than copper TSVs [159]. Therefore, we assume the diameter for W TSV is $3\mu m$ and the resistance of Al TSV and W TSV are about $1.68X$ and $9.26X$ larger than that of Cu TSV, respectively. We adopt the activation energy for Cu, Al, and W that are reported as 0.81eV, 0.6eV, and 1.9eV [139, 160]. The current load remains the same for all three materials.

Figure 5.9 shows the EM lifetime distribution of TSV arrays with different filling materials. The result indicates that copper TSV arrays have the longest EM lifetime, whereas aluminum and tungsten TSVs have the similar EM behavior. However, the differences of three EM lifetimes are small. The resistance affects the current distribution within an array and the current density through each TSV. Tungsten TSVs have much larger resistance than copper TSVs due to the high aspect ratio. Therefore, even though tungsten TSVs has large activation energy, their EM MTTF is still smaller than copper TSVs. Similarly, the resistance of aluminum TSVs is larger than copper TSVs while the activation energy is smaller, which leads to the small EM MTTF. Note that from previous studies, tungsten TSVs have the worst EM behavior due to the weak atom diffusion path at the interface [160]. In this work, we focus on the current density analysis of filling material inside bulk TSV that it is not included in their study.

5.3.3 TSV Number and Size in Array

In the following set of experiments, we examine the EM lifetime sensitivity to three design parameters: TSV counts, TSV sizes, and current loads. First, we increase the number of TSVs with the same current load and then increase the current load to analyze the EM lifetime behavior. In addition, we observe the EM MTTF change with various TSV diameters under constant current load. Finally, considering the area overhead, we sustain the total TSV area budget and study the tradeoff between the TSV count and size.

With fixed current stress, when we increase the number of TSVs in the array, the EM lifetime can approximately be improved linearly as shown in Figure 5.10. As expected, when more TSVs are used in the array, the EM MTTF is increased because smaller current density is assigned to each TSV. However, when the current load is increased to two or even four times larger, the benefit of adding more TSVs is reduced. Therefore, in the real design, we should carefully plan the number of TSV arrays in the whole circuit to guarantee that current load on each TSV array is not too large that may degrade the circuit performance.

To examine the sensitivity of EM lifetime to the TSV size, we build different TSV arrays with the diameter varies from $2\mu m$ to $10\mu m$. The results of a 2×2 TSV array are shown in Figure 5.11. Once the current load and TSV length are fixed, the larger TSV diameter results in the significant growth of the EM MTTF with the growth trend shown in the dot line. Increasing the TSV diameter can help alleviate the EM degradation because larger TSV diameter can reduce both the resistance and current density. Nevertheless, it introduces additional area overhead as the side effect. Moreover, the MTTF grows nearly exponentially with increased TSV size at the beginning, however, the improvement is smaller with continuously sizing.

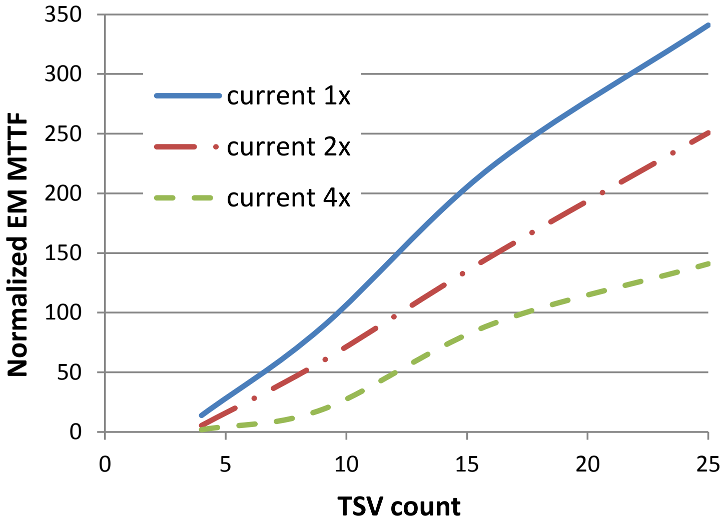

Figure 5.10. TSV array EM MTTF with different TSV counts in the array under three different current stresses.

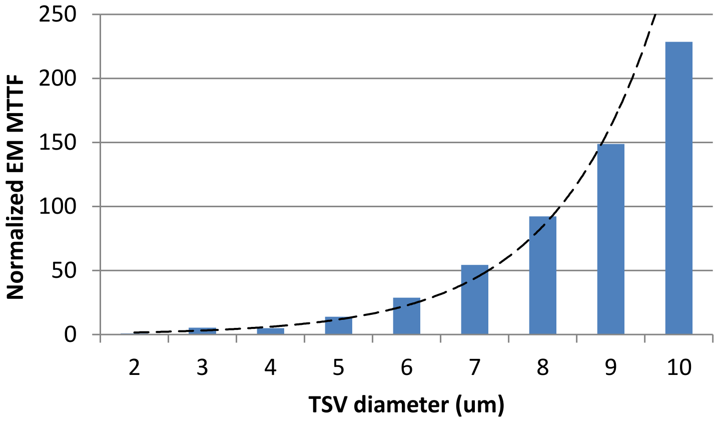

Figure 5.11. TSV array EM MTTF with different diameters under constant current load.

From the above analysis, the results show that increasing the the number and size of TSV can both help improve the EM lifetime as expected. The area overhead, however, prevents the aggressively sizing of TSV arrays. Normally, the overall area budget for TSVs is determined at the early stage of a design cycle. With the fixed area budget, we should make the trade-off between the size and number of TSVs to achieve the desired EM lifetime. The TSV size should be reduced when more TSVs are inserted. In the experiment, we increase the number of TSVs in the array from 4 to 25, with predetermined current load and $314\mu m^2$ TSV area. Table 5.1

Table 5.1. EM MTTF with fixed area budget.

TSV count	**TSV diameter (μm)**	**EM MTTF (a. u.)**
2×2	10	221.20
3×3	6.67	374.37
4×4	5	219.93
5×5	4	146.52

shows that as we increase the number of TSVs from 4 to 9, the EM lifetime is improved at first. The EM lifetime, however, is shortened if we further increase the number of TSVs to 25. The aggressively decreased TSV size becomes the dominant factor and suppresses the further lifetime extension. With the given configuration, the turning point appears when 3×3 TSV array is employed with TSV diameter equivalent to $6.67\mu m$.

In summary, the effective TSV area can be obtained from increasing the number of TSVs or TSV diameter. Compared with the methods of increasing TSV count and TSV diameter, we find that using a large TSV is the more efficient way to augment the EM lifetime during design time, because the current density is inversely proportional to the square of TSV diameter when the current load is fixed. Furthermore, the large TSV can help to reduce the IR drop between tiers and act as thermal vias for heat dissipation. The experiment results imply that even though a smaller TSV can alleviate the parasitics, for P/G TSVs, a larger TSV is preferred considering EM degradation.

5.4 Case Study

In this section, we perform a case study with the extracted power grid resistance and current load information from the circuit in IBM power grid benchmark [161]. We select four blocks with different power characteristics and duplicate the blocks to form a two layer stacking. A 5×5 TSV array is inserted for power supply. The EM lifetimes of the TSV array with both even and uneven current distributions are evaluated with the proposed analytical framework. The total execution time is less than 1 minute. The results show that the calculated EM MTTF under uneven current distribution is only about 39.45% of the MTTF when current is evenly distributed. The results indicate that the EM lifetime can be extremely overestimated if even current distribution is assumed. As the uneven current distribution is absent in all previous works, the TSV arrays in those works would have unexpected reliability issues during circuit operation.

Considering the uneven current distribution, we increase the TSV number in the array to achieve the same MTTF as expectation with even current distribution. Based on the experiment results, only when we increase the number of TSVs from 25 to 36 (a 6×6 TSV array), the desired MTTF can be reached. The introduced additional area overhead should be taken into consideration during design time.

5.5 Summary

Electromigration is an important reliability issue in nanoscale VLSI circuit designs. In this work, we propose an analysis framework for EM lifetime of power supply TSV arrays, focusing on the impact of current distribution among the array. The results show that if the current distribution is not evaluated correctly, the estimated EM lifetime is misleading, resulting in unexpected early failure of TSV arrays. The sensitivity studies of design parameters are discussed in the paper to show their impacts on EM lifetime. In general, TSV diameter is considered to be the most effective design metric to prolong the EM lifetime.

Chapter 6

Crosstalk Minimization in TSV Arrays

The signal integrity issues in TSVs become the major challenges in 3D designs [142, 162][1]. Studies show that the coupling problem is not negligible in TSVs because of the relatively large diameter, which results in TSV-to-TSV coupling, TSV-to-device coupling, and TSV landing pad to wire/device coupling. In these coupling effects, the TSV-to-TSV coupling attracts lots of research interests because of the relatively large coupling noise and the lack of sufficient mitigation methods. Elaborated analysis and modeling methods are provided for the TSV-to-TSV coupling [85, 162–164]. They claim that TSV-to-TSV crosstalk problem is an important signal reliability issue and should be taken into consideration during design phase.

Several crosstalk minimization techniques have been proposed in 2D designs, such as active shielding [165], data coding [166–168], and wire spacing [169]. Other techniques to migrate the crosstalk delay with negligible area overhead are proposed, such as alleviating the crosstalk effect through transmission cycle tuning [170] by observing that not all the transmissions result in large crosstalk delay. Nevertheless, these techniques can not be directly applied in 3D designs. The additional dimension results in significant difference in crosstalk problem analysis between 2D and 3D designs. In most 2D cases, the coupled wires are usually considered placed in the same planar and each victim has only at most two aggressors, however, each victim TSV is surrounded by at most eight aggressors in 3D and the crosstalk noises come from the coupling capacitance network. Consequently, the crosstalk analysis and elimination become complex in 3D designs.

---

[1] This work is published as "3DLAT: TSV-based 3D ICs crosstalk minimization utilizing less adjacent transition code" on ASPDAC 2014.

In this chapter, we propose a coding mechanism called ω-Less Adjacent Transition (LAT) to reduce the crosstalk delay and transmission power in a TSV array. This mechanism encodes the input data to a codeword which contains limited number of 1s (indicated by ω) in every 3×3 TSV array. In the analysis, we show that by applying our coding mechanism, the crosstalk delay can be reduced by 38% and the power consumption overhead is minimized, when we constrain at most four 1s in each 3×3 array (ω=4). Further delay reduction can be obtained with a smaller value of ω with the cost of area overhead.

6.1 Preliminaries

In this section, we first introduce the capacitance crosstalk model that is used in our mechanism. According to the crosstalk model, we classify the crosstalk into 10 classes for the convenience of analysis in the following sections. After that, the previous studies about crosstalk reduction and elimination in both 2D and 3D designs are briefly introduced.

6.1.1 Crosstalk in TSV Array

The coupling capacitance network among adjacent TSVs is shown in Figure 6.1. In this model, the middle TSV i is the victim and proximate eight TSVs are aggressors. Due to the different distance, we use C_d and C_c to represent the coupling capacitance of diagonal TSVs and vertical/horizontal TSVs, respectively. C_L is the capacitance between TSV and bulk silicon, also known as the self-capacitance. With the notation above, the approximate signal delay considering the capacitance crosstalk can be expressed in the following equation [171, 172]:

$$\tau_i(\alpha) = -RC_L(1 + \lambda_1(\delta_{i,i-3} + \delta_{i,i-1} + \delta_{i,i+1} + \delta_{i,i+3}) +$$

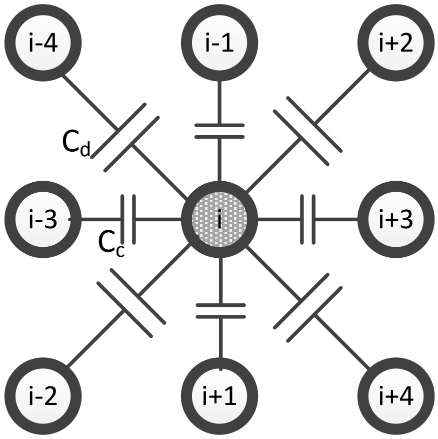

Figure 6.1. The capacitance crosstalk model in a 3×3 TSV array.

$$\lambda_2(\delta_{i,i-4} + \delta_{i,i-2} + \delta_{i,i+2} + \delta_{i,i+4})) \tag{6.1}$$

where

$$\begin{cases} \lambda_1 = \frac{C_c}{C_L} \\ \lambda_2 = \frac{C_d}{C_L} \\ \Delta V_i = V_i(t^+) - V_i(t^-) \\ \delta_{i,k} = abs(\frac{\Delta V_i - \Delta V_k}{V_{dd}}) \end{cases}$$

In the above equations, $V_i(t^+)$ and $V_i(t^-)$ denote the voltages before and after the transition. The value of $\delta_{i,j}$ can only be 0, 1, or 2 to represent the relative signal transition direction between victim and aggressor. For example, if two signals in TSV i and k are switching in the opposite directions (i.e. $0 \rightarrow V_{DD}$ and $V_{DD} \rightarrow 0$), $\delta_{i,j}$ equals to 2 because $\Delta V_i = V_{DD}$ and $\Delta V_k = -V_{DD}$. If two signals are switching in the same direction, $\delta_{i,j}$ takes the value of 0. Note that Equation 6.1 is applied when a transition happens on victim TSV i, otherwise, there is no signal delay since signal i stays unchanged. Assuming the ratio between C_d and C_c is 1/4 [173], the effective crosstalk capacitance $C_{eff,i}$ on victim TSV i is defined as follows:

$$\begin{aligned} C_{eff,i} = C_L(1 + \lambda_1(\delta_{i,i-3} + \delta_{i,i-1} + \delta_{i,i+1} + \delta_{i,i+3}) + \\ \frac{\lambda_1}{4}(\delta_{i,i-4} + \delta_{i,i-2} + \delta_{i,i+2} + \delta_{i,i+4})) \end{aligned} \tag{6.2}$$

Based on the effective coupling capacitance, we classify the crosstalk into 10 categories. In addition to **0C** to **8C** classes that have been defined in [173], we take the diagonal TSVs into consideration and extend the crosstalk classes to **9C** and **10C**. In brief, the classification can be done in the following steps:

- Initialize crosstalk class to **0C**;
- Determine the transition direction of TSV i;
- Determine the accumulated $\delta_{i,k}$ for each vertical/horizontal TSV according to the definition and increase the crosstalk class by $\sum \delta_{i,k}\mathbf{C}$;
- Determine the accumulated $\delta_{i,d}$ for each diagonal TSV, if the value exceeds 0.5, increase the crosstalk class by **1C**.

6.1.2 Related Work

Plenty of research have been done to minimize or eliminate the crosstalk delay in 2D design to close the increasing delay gap between interconnect and transistor. Generally, there are mainly two types of methods to handle the crosstalk problem: static methods and data coding (CAC) [165–169]. Static methods include data shielding and wire spacing in design time. The benefit of static methods is that data input doesn't need to be changed. However, large overhead is introduced. Coding schemes try to minimize crosstalk by coding the data input to avoid patterns that cause large crosstalk delay during data transmission. These methods have been proposed and their effectiveness is proved in the 2D regime.

Even though the crosstalk handling methods are mature in 2D designs, they can't be directly applied on 3D designs. Unlike in the 2D bus, where usually only two adjacent wires are considered as aggressors, the additional dimension in 3D interconnect increases the complexity in crosstalk minimization. Shielding and coding methods have been initially explored, focusing this unique feature of 3D designs.

The crosstalk problem and influence factors have been examined in 3D designs [85, 164, 174]. These studies prove that by increasing the pitch or inserting power/ground TSVs between signal TSVs, the crosstalk noise can be significantly reduced. Nevertheless, this kind of shielding method introduces large area overhead and increases the routing complexity during design time. ShieldUS [172] suggests to use data signals as shields to minimize the crosstalk during data transmission. This approach remaps the bits that are relatively stable as shields to separate the more active TSVs. The area overhead is negligible in this method. However, the worst case delay cannot be guaranteed and dynamic remapping circuitry introduces overhead. 3D Crosstalk avoidance code (CAC) is proposed and analyzed with different levels of crosstalk minimizations [173]. The benefit of such a coding scheme is that it can provide the desired level of crosstalk minimization, therefore, the transmission delay can be guaranteed. The disadvantage is that it requires significant extra interconnect resources and the crosstalk from diagonal TSVs is not considered.

In this work, we propose a coding mechanism for 3D ICs, called Less Adjacent Transmission. In addition to the consideration of diagonal TSV, the proposed approach can guarantee the crosstalk delay in different levels while reduce the power consumption overhead in data transmission.

6.2 3D LAT Coding Mechanism

In this section, the Less Adjacent Transition (LAT) coding mechanism is introduced. Since the LAT code is derived from 2D No Adjacent Transition (NAT) code [166], we briefly review the NAT coding scheme which is used on 2D design and can reduce the bus crosstalk delay. Through analytical analysis, we show that the NAT code is impractical to apply in 3D ICs. Then we propose the LAT code for 3D designs which can minimize the crosstalk delay in TSV arrays while maintain the coding overhead within reasonable range. Considering the non-negligible overhead, we further propose the overhead optimization schemes. Finally, the heuristic strategy of encoder and decoder (CODEC) designs are given.

6.2.1 Preliminaries in 2D NAT Code

In the coding scheme, the original input is defined as *dataword* and the data after coding is called *codeword*. The coding scheme is used to design a sequence of codeword that satisfy several constraints and map each dataword into the corresponding codeword. Due to the constraints, usually the codeword width is longer than that of the dataword, resulting in high coding overhead and potentially larger power consumption. In order to minimize both crosstalk and power, the NAT coding scheme is proposed [166]. In addition to the Transition Signaling and the Limited Weighted Code, NAT coding scheme constrains that no adjacent transition is allowed.

Transition Signaling defines that the transition takes place only when the input bit is a 1. For example, if the codeword is 00100, then the middle wire in the bus changes to its compliment value while others remain unchanged. The encoder and decoder can be simply constructed with XOR gates. The number of 1's in the dataword is called the data weight. Limited Weighted Code is a coding method that limits the maximum weight in the codeword. Therefore, the number of transitions is bounded by the maximum weight. Combining these two transmission techniques, the NAT coding mechanism restricts that no adjacent 1s are allowed in the codeword, therefore, crosstalk is eliminated because no adjacent transitions are allowed. Furthermore, power consumption with NAT is reduced because it is proportional to the maximum weight.

This coding scheme can be easily applied to a 2D bus since in crosstalk analysis, each wire usually only considers two adjacent wires. However, the adjacency concept in 3D designs has changed. Therefore, if we still apply the rule that no adjacent 1s are allowed in the codeword, it would mean that every 1 requires eight surrounding 0s. In consequence, the codeword overhead is extremely large. A Less Adjacent Transition (LAT) coding scheme is proposed to handle this unique 3D crosstalk challenge.

6.2.2 3D ω-LAT Code

The framework overview of signal transmission with LAT code is shown in Figure 6.2. The data input first goes through the LAT encoder to generate the corresponding codeword. Before transmission, the transition signaling encoder which is constructed with XOR gates is applied. At the receiver side, the decoding steps (transition signaling decoder and LAT decoder) are performed to generate the final data output.

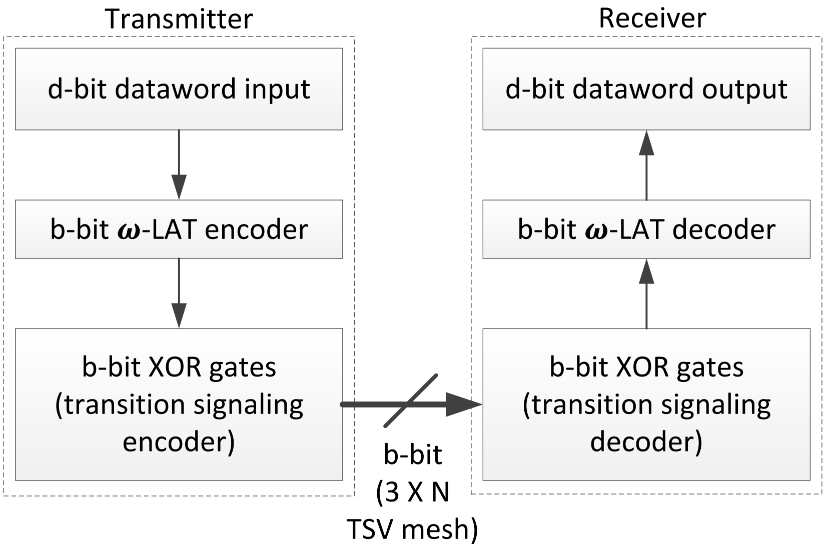
Transmitter
d-bit dataword input
b-bit ω-LAT encoder
b-bit XOR gates (transition signaling encoder)
b-bit (3 X N TSV mesh)
Receiver
d-bit dataword output
b-bit ω-LAT decoder
b-bit XOR gates (transition signaling decoder)

Figure 6.2. Framework overview of signal transmission with LAT coding scheme.

Before introducing the LAT coding scheme, we need to clarify two concepts that are used in the analysis.

Definition For each single TSV i, all the surrounding eight TSVs, including diagonal TSVs and horizontal/vertical TSVs are called i's adjacent TSVs. The victim TSV and its eight adjacent TSVs construct a 3×3 TSV array, which is called TSV subarray throughout this paper.

As illustrated in previous section, it is impossible to have no adjacent transitions in a 3D TSV array. Therefore, we relax this requirement and propose a ω-LAT code. ω represents the maximum weight for every TSV subarray, that is, every TSV and its eight adjacencies. By restricting the maximum weight, we can minimize the worst case crosstalk delay within each subarray as shown in the following.

Lemma 1. *For each ω-LAT code when $\omega \leq 5$, the crosstalk class of the transmission will not exceed* $(\omega - 1)$***2C**.

Proof. From the definition, we can see that crosstalk class is determined from $\delta_{i,j}$. In a TSV subarray, if the transition direction of the middle TSV i is opposite to the transition of its direct neighbor j, then $\delta_{i,j}$ equals to 2. Therefore, for a ω-LAT code, the worst case of crosstalk delay happens when the signal for middle TSV is 1 and signals for $\omega - 1$ of its horizontal/vertical neighbors are 1s, meanwhile, the transition directions of these TSVs are in opposite. In this case, the crosstalk is maximum and the class is **($\omega - 1$)*2C**. Since each subarray follows this rule, this conclusion can be easily applied to a N×N TSV array which contains multiple subarrays. □

Next, we will introduce how to calculate the code cardinality of a ω-LAT code that can satisfy the one-to-one mapping between codeword and dataword. Due to the calculation complexity, we only show the analytical analysis of a 3×N TSV array. The terms that are used in the following analysis are shown below:

d = dataword bitwidth

N = the number of TSV columns in the codeword, therefore, the codeword bitwidth is $3 * N$

α_c = the weight of column c, since we only consider 3 rows, α_c cannot exceed 3

$T(\beta, c)$ = the number of codewords (code cardinality) when the TSV array has c columns and the weight of each subarray is exactly β

$T_\omega(c)$ = the number of codewords with c TSV columns when the maximum weight of each subarray is ω

Based on the definitions and known conditions, the code cardinality with maximum weight ω can be calculated as follows:

$$T_\omega(N) = \sum_{\forall c, \alpha_c + \alpha_{c+1} + \alpha_{c+2} \leq \omega} \prod_{c=0}^{N} \binom{3}{\alpha_c} \tag{6.3}$$

In this equation, we assume that each column has weight α_c, and the weight in consecutive three columns cannot exceed ω. When N is small, the value of $T_\omega(N)$ can be generated with enumeration. However, when size N increases, it is infeasible to solve this equation within polynomial time. Alternatively, we consider the feasible lower bound of code cardinality which can be described with the following equation:

$$T_\omega(N) \geq \sum_{\forall c, \alpha_c + \alpha_{c+1} + \alpha_{c+2} = 0}^{\omega} \prod_{c=0}^{N} \binom{3}{\alpha_c} = \sum_{\beta=0}^{\omega} T(\beta, N) \tag{6.4}$$

where β is the total weight for any consecutive three columns as defined. The above equation can calculate the code cardinality's lower bound because it sacrifices the coding flexibility.

In Equation 6.4, we constrain that the total weight in any consecutive three columns (consecutive subarray) are the same to provide the lower bound. The calculation can be done in inductive procedure. Since each subarray contains the same weight, the equation $\alpha_{N-3} + \alpha_{N-2} + \alpha_{N-1} = \alpha_{N-2} + \alpha_{N-1} + \alpha_N = \beta$ is held. Therefore, the weight in column N should be the same as in column $N-3$, meaning the weights in every three columns are the same. The following equation is constructed assuming N is an integer multiple of 3:

$$\begin{aligned}
T(\beta, N) &= \sum_{\alpha_1+\alpha_2+\alpha_3=\beta} \left[\binom{3}{\alpha_1}\binom{3}{\alpha_2}\binom{3}{\alpha_3}\right]^{\frac{N}{3}} \\
T(\beta, N+1) &= \sum_{\alpha_1+\alpha_2+\alpha_3=\beta} \left[\binom{3}{\alpha_1}\binom{3}{\alpha_2}\binom{3}{\alpha_3}\right]^{\frac{N}{3}} \binom{3}{\alpha_1} \\
T(\beta, N+2) &= \sum_{\alpha_1+\alpha_2+\alpha_3=\beta} \left[\binom{3}{\alpha_1}\binom{3}{\alpha_2}\binom{3}{\alpha_3}\right]^{\frac{N}{3}} \binom{3}{\alpha_1}\binom{3}{\alpha_2} \qquad (6.5)
\end{aligned}$$

Note that the previous equations can be used to derive the codeword count when $N \geq 3$. For N smaller than 3, enumeration is applied to get the code cardinality. Given the value of β, all the possible combinations of α_1, α_2, and α_3 subjected to $\alpha_1 + \alpha_2 + \alpha_3 = \beta$ are enumerated and the corresponding code cardinality is calculated.

In order to satisfy the mapping between dataword and codeword, we should find the minimal N such that $T_\omega(N) \geq 2^d$. Therefore, the overhead can be calculated from $\frac{3*N-d}{d}$. The coding overhead with respect to the input data bitwidth is shown in Figure 6.3.

When the ω is reduced, the overhead increases to achieve lower crosstalk delay. The upper bounds of ω =2, 3, and 4 are 190%, 100%, and 90%, respectively. The overhead of ω=2, which means the maximum crosstalk in transmission is 2C, is significantly larger than the other two cases. When the data bitwidth is smaller than 15, the overhead grows proportionally with the increased data input. After

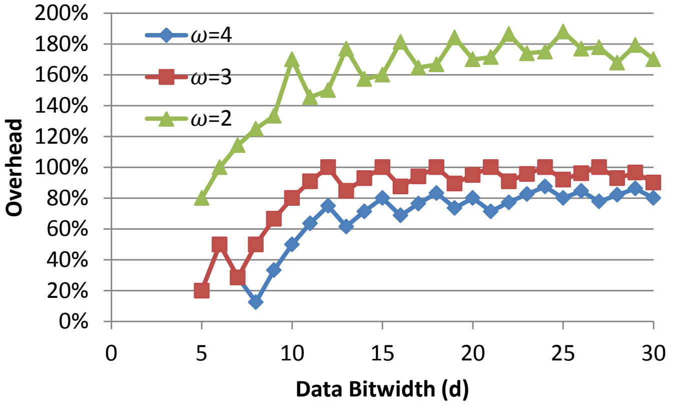

Figure 6.3. The overhead caused by redundant TSVs with different ω. The horizontal axis is the bitwidth of dataword and the vertical axis is the overhead percentage.

that, it varies within a small range and approaches the asymptotic upper bound. Note that the calculated cardinality is the lower bound of ω-LAT code, therefore, the overhead is the upper limit. Even though the overhead in LAT is larger than 3D CAC [173] with 4C and 6C crosstalk reduction, it is significantly smaller when the crosstalk is aggressively reduced to 2C. Moreover, the estimated overhead in 3D CAC is a lower bound while our overhead is the upper bound.

6.2.3 LAT Code Optimization

In order to reduce the code overhead, the first simple technique we can apply is bus inverting [175]. A global weight detector is used to determine if the dataword's weight exceeds $d/2$. If the weight is larger than $d/2$, we change the input data to its complement. The bus inverting bit is set at the transmitter side which will indicate the reverse operation at the receiver side. Therefore, the codeword cardinality only needs to be larger than $2^{(d-1)}$. The weight detector can be implemented with hamming distance circuits by comparing the hamming distance between data input and all 0's.

The main purpose of LAT coding scheme is to restrict the weight in the TSV subarray for crosstalk minimization. Therefore, if the dataword doesn't violate this restriction, we don't need to perform the coding. By excluding the qualified dataword, we can reduce the number of inputs that needs to be encoded, and further reduce the codeword length and overhead. In this case, a local weight

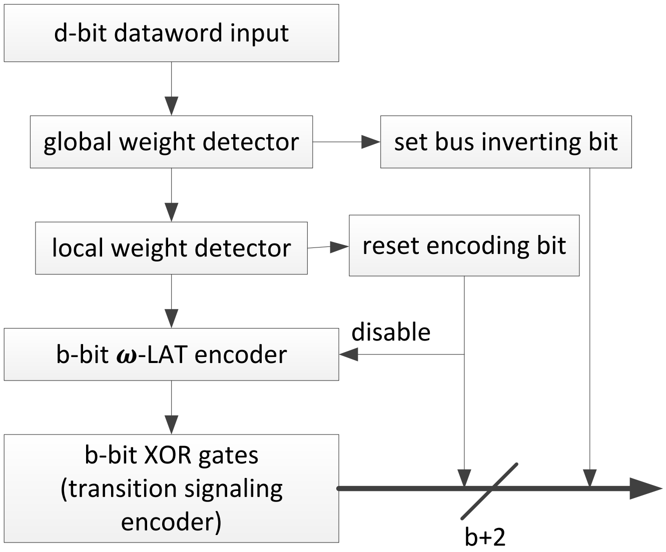
d-bit dataword input
global weight detector
set bus inverting bit
local weight detector
reset encoding bit
disable
b-bit ω-LAT encoder
b-bit XOR gates (transition signaling encoder)
b+2

Figure 6.4. The overview of signal transmission procedure after overhead optimization.

detector is needed to determine if every TSV subarray has its weight smaller than ω. If the dataword doesn't need the encoding, the encoding bit is reset and the encoder is disabled. The data input is directly sent to the receiver through TSV arrays. In total, $d/3 - 2$ number of 9-bit weight detectors are required.

The block diagram of the optimized coding mechanism at the transmitter side is shown in Figure 6.4. In the optimization scheme, two extra bits are needed: a bus inverting bit and an encoding bit. The receiver side performs decoding and bus inverting based on these two signal bits accordingly. According to the previous analysis, the codeword length is decided by finding the minimal N to satisfy the following condition:

$$T_\omega(N) \geq 2^{(d-1)} - T_\omega(d/3) \tag{6.6}$$

Equation 6.4 is used to calculate the lower bound of $T_\omega(N)$. Table 6.1 show the coding overhead comparison of LAT code with and without optimization when $\omega = 4$. From the results, for small input size, large overhead saving can be obtained. For example, if we need to encode a 5-bit data input, after optimization, only 3 TSVs would be needed. Considering the two added signal bits, there is no overhead. Nevertheless, as the data input size increases, the overhead saving is reduced. It is because when the input size increases, the percentage of qualified patterns (ratio

Table 6.1. Coding overhead comparison results of with and without optimization.

Data Width	**Optimized**		**Original**		**Reduced Ratio**	**Reduced Overhead**
	column	overhead (%)	column	overhead (%)	($\frac{T_\omega(d/3)}{2^d}$)	(%)
5	1	-40	2	20	25	60
10	3	-10	5	50	25	60
15	8	60	9	80	4.04	20
20	11	65	12	80	0.38	15
25	15	80	15	80	0.02	0

of $T_\omega(d/3)$ to 2^d) is reduced. Meanwhile, in the optimization scheme, the size of both global and local weight detectors grows linearly with increasing dataword length. The corresponding power consumption and detection delay increase with longer input, therefore, the power and delay saving from crosstalk minimization is sacrifices. The design tradeoff should be carefully considered to obtain the optimal power and performance.

6.2.4 Heuristic LAT CODEC Design

In LAT design, each subarray should contain weight no more than ω. The codeword cardinality changes with different ω values, therefore, it is hard to design a universal encoder and decoder for every possible ω and data bitwidth. The CODEC can be designed only after the codeword length is determined through data bitwidth and ω during design time. In general, two levels of comparators are needed in the encoder. The first level is used to decide the weight in each TSV subarray and the second level is used to select the combination of α_1 to α_3. In the following analysis, we assume ω=4, data bitwidth equals to 16, and a data input value of 1024 as an example to explain the CODEC design.

Based on the configuration, the required TSV column to encode 16-bit data is 9 and the total codeword bitwidth is 27. The codeword cardinalities with weights equal to 0, 1, 2, 3, 4 are 1, 81, 2268, 24060, 61398, respectively. Therefore, we need five comparators on the first level, where each has a value 1, 82, 2350, 26410, and 87808, respectively. These comparators operate in parallel to reduce timing overhead. Data 1024 is within the range of 82 to 2350, therefore, the weight in each subarray should be 2.

The possible number of combinations of α_1 to α_3 is fixed when the subarray weight is known. For maximum weight of 4, 12 combinations are generated, therefore in the second level, 12 comparators are needed. For each combination, we can calculate the codeword cardinality and determine the α_1 to α_3 combination. For example, the input 1024 should be encoded with $\alpha_1 = 1$, $\alpha_2 = 0$ and $\alpha_3 = 1$.

Next, we need to decide the row position of 1s. The total weight is 6 from the value of α. For column containing 1s, we define a coefficient k, and k is 0, 1, or 2 when the 1-valued bit is on row 0, 1, or 2. Therefore, the data value can be represented in $k_0 * 3^0 + k_1 * 3^1 + ... + k_5 * 3^5$. In the last step, we calculate the value of these coefficients and generate the codeword. For example, the coefficients of k_0 to k_5 for data 1024 are 0, 2, 2, 2, 1, and 0, respectively.

The 16-bit comparator is implemented and synthesized with Nangate 45nm library using Synopsys Design Compiler. The total area consumed by the two-level comparators is about $4264 \mu m^2$ while the signal delay of each level is about $0.86ns$. Pipeline design can be used to reduce the timing overhead from CODEC.

At the decoder side, we first use one weight detector to examine the weight of one TSV subarray. After that, three weight detectors are applied on each column in the subarray to determine α_1 to α_3 combination. And based on the row position of the 1-valued bit, we can get the coefficient and the data input.

6.3 Evaluation

In this section, we perform the evaluation of our proposed ω-LAT coding scheme and compare our scheme with two 3D crosstalk elimination mechanisms: ShieldUS [172] and 3D CAC code [173]. First, we introduce the power model to analyze the interconnect power consumption of these three mechanisms. After that, the crosstalk reduction analysis and effectiveness on real benchmark traces are shown.

6.3.1 Interconnect Power Analysis

Because of the capacitive crosstalk in interconnects, the dynamic power consumption of each TSV P_k mainly comes from two parts: the switching power P_k^s caused by wire transition and the coupling power P_k^c caused by inter-wire transition [176]. The total power consumption can be obtained by $P = \sum_{k=1}^{3N} P_k^s + \sum_{k=1}^{3N-1} P_k^c$. In order to avoid redundant summation of same item, we consider three adjacencies, the south, east and southeast of each TSV in the crosstalk power. Similar to the analysis in [166], the equation of P_k^s is shown as follows:

$$P_k^s = \frac{1}{2} C_L V_{DD}^2 * Pr(trans) \tag{6.7}$$

where $Pr(trans)$ represents the probability of transition in TSV k. Due to the increased number of neighbors, the value of P_k^c can be calculated from the following equation:

$$\begin{aligned} P_k^c &= C_c V_{DD}^2 * Pr(V_k(t^+) \neq V_{k+1}(t^+)) * E_t(k, k+1) \\ &+ C_c V_{DD}^2 * Pr(V_k(t^+) \neq V_{k+3}(t^+)) * E_t(k, k+3) \\ &+ C_d V_{DD}^2 * Pr(V_k(t^+) \neq V_{k+4}(t^+)) * E_t(k, k+4) \end{aligned} \tag{6.8}$$

where $k+1$, $k+3$, and $k+4$ denote the south, east and southeast adjacencies of TSV k. $Pr(V_k(t^+) \neq V_{k+1}(t^+))$ represents the probability of different voltage between TSV k and its neighbor after transition. $E_tc(k, k+1)$ is the expected number of transitions in $(V_k(t^-), V_{k+1}(t^-)) \rightarrow (V_k(t^+), V_{k+1}(t^+))$ when $V_k(t^+) \neq V_{k+1}(t^+)$.

Compared to uncoded data transmission, the coding scheme changes the probability of transition and the expected value of transition count between two coupled TSVs. Table 6.2 shows the parameter values when the input bitwidth is 15 in uncoded cases, ShieldUS, 3D CAC, and the proposed ω-LAT schemes. Due to space limitation, the difference between boundary and middle TSVs are not listed. For fair comparison, we choose ω as 4 and the 3D 6C CAC, which means that the maximum crosstalk is 6C. Since 3D CAC doesn't consider the diagonal TSVs, we hereby assume that the parameters in diagonal TSVs are the same as in direct neighbors.

For uncoded and ShieldUS, the input data doesn't have any constraints, therefore, $Pr(trans)$ and $Pr(V_k(t^+) \neq V_{k+1}(t^+))$ are all $\frac{1}{2}$. The transition count between two TSVs is hard to predict in ShieldUS since it is still highly dependent on the input data. Nevertheless, the expected transition should be smaller than uncoded cases due to the shielding effect.

For 6C CAC, we implement and calculate the valid patterns count to code a 16

Table 6.2. The power consumption related parameters comparison between uncoded, ShieldUS, 3D 6C CAC, and 3-LAT schemes.

code	$Pr(trans)$	$Pr(V_k(t^+) \neq V_{k+1}(t^+))$	$E_tc(k,k+1)$
uncoded	0.5	0.5	1
ShieldUS	0.5	0.5	≤ 1
6C CAC	0.5	0.367	1
4-LAT	0.4079	0.5	0.8159

bit data. The $Pr(V_k(t^+) \neq V_{k+1}(t^+))$ is smaller than 0.5 in 6C CAC because this coding scheme only contains the valid pattern. Valid pattern is determined by the complemental value count in the four neighbors. For example, the definition of a valid 2C pattern is that for the central TSV, only one out of four direct neighbors can has its complement value. Therefore, the probability that two nearby bits have different values reduces from $\frac{1}{2}$ to $\frac{1}{4}$.

In 4-LAT cases, since transition signaling is applied, therefore, transition happens only when the encoded data is 1. By limiting the weight in each subarray, we can reduce the probability of transition. $Pr(trans)$ can be calculated as $\sum_{\beta=0}^{\omega} \frac{\beta}{9} * \frac{T(\beta,N)}{T_\omega(N)}$. When the value of ω reduces, the transition probability also reduces. If other conditions are the same and ω changes from 4 to 3, the probability of transition reduces from 0.4079 to 0.3251.

Note that bit 1 represents transition in LAT, not the voltage value of the corresponding wire, while in Equation 6.8, V_k is the voltage value. We assume in the initial state, $Pr(V_k(t^+) \neq V_{k+1}(t^+))$ is $\frac{1}{2}$, then after the transition, the probability of inequality can be calculated considering two cases: the two initial voltages are the same, and one goes through a transition; the two initial voltages are different, both go through transitions or remain unchanged. In each case, $Pr(trans)$ is used for transition probability. Similarly, the parameter $E_tc(k,k+1)$ can be expressed as the expected number of 1s in TSV k and $k+1$.

When the parameter λ_1 is set as 5.54 [172], for single TSV, the power consumption is $8.56C_LV_{DD}^2$ in uncoded cases and $6.98C_LV_{DD}^2$ in 4-LAT cases, respectively. Due to the TSV overhead after coding, the total power consumption of 4-LAT is slightly larger than uncoded cases, however, the average power consumption for each TSV is 18.46% smaller than uncoded cases.

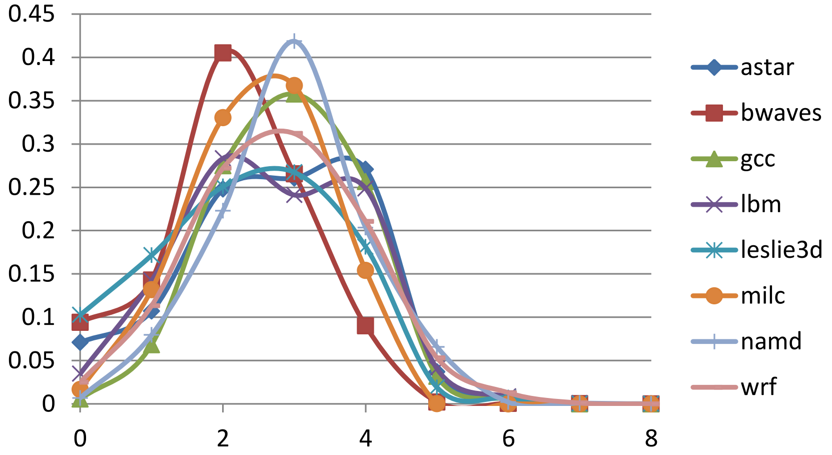

Figure 6.5. The benchmark crosstalk characteristic in 3D cases.

6.3.2 Crosstalk Delay Analysis

In the crosstalk minimization evaluation, we use the extracted data trace from benchmark SPEC2006 and simulate the transmission time of ShieldUS, 4C CAC, and 3-LAT. The architecture simulator GEM5 is used for memory data trace extraction. Each benchmark is executed with four cores and the Ruby memory model is used to connect cores to memory. The 3D crosstalk classification of the data trace is analyzed with our implemented crosstalk analyzer and the results are shown in Figure 6.5. Most of the data transmissions fall into the range of 2C to 4C. The data bitwidth is 64 and we divide every 16 bits into one group. Therefore, for every group, we use 4×4 TSVs in both uncoded and ShieldUS cases, while the TSV arrangements are 3×9 and 3×10 in 4C CAC and 3-LAT, respectively.

When λ_1 is set as 5.54, the transmission delays in 4C CAC and 3-LAT code are the same as $23.16RC_L$. For fair comparison, in ShieldUS, we set the delay time of 4C as one clock cycle and multiple cycle transmission method is used. The sampling interval for ShieldUS is 100 transmissions. For uncoded cases, the transmission cycle is determined by the longest crosstalk delay (**10C** class), which is $56.4RC_L$. We also simulate the transmission delay in the ideal case, which means the clock cycle is flexible and determined only by the crosstalk delay in each transmission.

Figure 6.6 shows the average benchmark data transmission delay of ShieldUS, 4C CAC/3-LAT, and ideal cases normalized to the uncoded case. Based on the

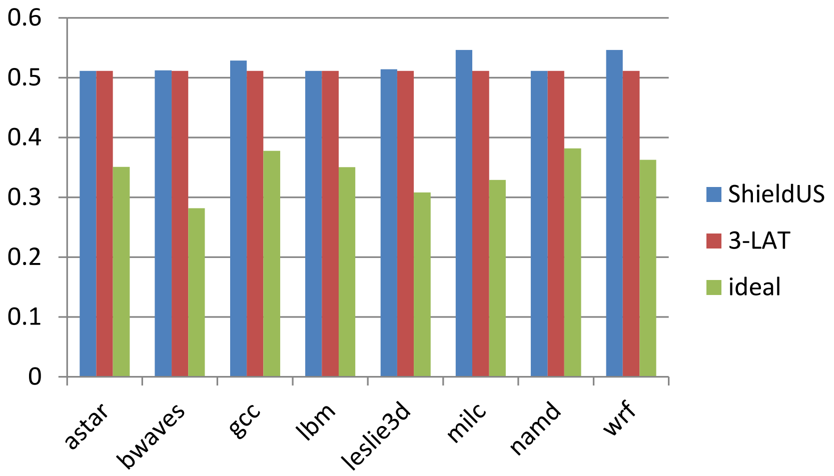

Figure 6.6. The transmission time results of ideal case, ShieldUS, and 6C crosstalk minimization schemes normalized to uncoded case.

benchmark characteristics, the most transmissions are in the range of 2C to 4C, therefore, the performance of ShieldUS is close to the 4C crosstalk minimization schemes (4C CAC and 3-LAT). The 4C crosstalk minimization schemes can guarantee the transmission time, however, ShieldUS can only reduce the delay when plenty of data bits are unchanged with no transmission time guaranteed. Moreover, from the experiment, we find that ShieldUS sometimes increase the transmission crosstalk.

6.4 Summary

Due to the complexity in 3D capacitive crosstalk minimization, we propose a novel ω-LAT coding scheme to minimize both the crosstalk and interconnect power consumption overhead in vertical interconnects. Combining transition signaling and adjacent weight limitation, the LAT coding can reduce the maximum capacitive crosstalk to 6C, 4C, or 2C when ω is 4, 3, or 2. An optimized mechanism is also introduced to reduce the codeword overhead. Compared with uncoded cases, our proposed coding scheme can achieve 38% and 58.9% interconnect delay improvement with ω equal to 4 and 3, respectively. The LAT code can reduce crosstalk to 2C with affordable overhead compared to 3D crosstalk avoidance code.

Chapter 7

Thermo-Mechanical Stresses Management in 3D ICs

As illustrated in Chapter 1, the increased power density and decreased chip footprint induce thermal related issues that have been observed as major barriers in 3D IC designs [177–179][1]. In addition, the Coefficient of Thermal Expansion (CTE) mismatch and thermal expansion directions can adversely affect TSV performance or even cause cracks in the entire die, which exacerbates the reliability of 3D ICs. Due to various design parameters (wirelength, chip area, etc.), only use design time solution to handle thermal stress is not adequate since the desired circuit placement may not be achieved. Moreover, the run-time operation influences on-chip temperature distribution, which cannot be captured during the design-time. To this end, 3D IC designs should carefully take into account the aforementioned thermal challenges in both design-time and run-time. Unfortunately, little work has been done to alleviate the challenge through these two stages. As a result, in this study we propose a two-stage, design- and run-time solution to this problem.

7.1 Background and Related Work

Stacked chips on 3D architecture increase the packaging density and thermal resistances, which results in higher on-chip temperatures. Extensive research has been done by focusing on the 3D thermal modeling, analysis [180–182], and thermal-aware design methodology [30,32] to manage the on-chip thermal issues of 3D ICs. However, the TSV lateral thermal blockage effect and thermomechanical stress weren't considered. Moreover, the view of using signal TSVs as thermal vias to build vertical heat dissipation path, which in turn results in increased thermal load on TSVs as well as thermomechanical stresses, and thus weakens the reliability. On the other hand, prior work on analyzing the mechanical stresses in 3D ICs [183,184] only consider the static stress management by adjusting TSV keep-out zone size, TSV placement, or TSV structure. Distinguished from previous work, this work not only accounts for the static (design-time) management of TSV thermal stress and thermal load but also takes into account the run-time TSV stress analysis and management.

---

[1] This work is published as ”Thermomechanical stress-aware management for 3D IC designs” on DATE 2013.

7.1.1 Analysis of TSV Thermal Stress

In 3D IC fabrication, copper (Cu) is usually used as TSV filling material that has more than **five** times larger CTE than silicon. The CTE mismatch between TSV and silicon substrate in turn introduces mechanical stresses that can lead to high probability of die cracking and interfacial delamination [185,186].

To minimize the thermomechanical stresses, the placement of TSV farms can be optimized during design time. Therefore, the corresponding analysis on the thermal stresses around TSVs is critical to the solution. There have been several work [158] [187] targeting on the thermal stresses analysis showing that the stress field in TSVs is uniform and can be represented by radial, circumferential, and axial stresses. The stresses can be expressed as followings:

$$\sigma_r = \sigma_\theta = \frac{-E(\alpha_{tsv} - \alpha_{si})T_{tsv}}{2 - 2\upsilon}, \sigma_z = 2\sigma_\theta \tag{7.1}$$

where σ_r, σ_θ, and σ_z are radial, circumferential, and axial stresses, respectively. α_{tsv} is the CTE of TSVs and α_{si} represents the CTE of silicon. T_{tsv} is the thermal load on TSV, E is the Young's modulus and υ is the Poisson's ratio[2].

From the equation above, the stresses in TSV are proportional to the thermal load and CTE mismatch. When the material and diameter of TSVs are determined, the only variable is T_{tsv}, the thermal load of TSVs. Therefore, the proposed design-time thermal management scheme alleviates thermomechanical stresses by reducing the thermal load on TSVs.

---

[2] In this formula, the difference of elastic between materials is omitted for simplicity.

7.1.2 TSV Lateral Thermal Blockage Effect

TSVs thermal load estimation during design-time is usually based on accurate thermal modeling of TSVs. Both vertical high thermal conductivity and lateral thermal blockage effect [182] are considered in our TSV model for more accurate temperature modeling.

TSV lateral thermal blockage effect is found due to the due to the unequal thermal conductivities in vertical and horizontal directions. Specifically, the liner layer between TSV filling material and silicon substrate has higher thermal resistance. Therefore, dense TSV farms can improve the thermal dissipation on the vertical direction, on the other hand, the lateral thermal dissipation path is blocked by liner layers. To capture both vertical and horizontal thermal characteristics, the TSV thermal resistance can be modeled as:

$$R_{dir} = \frac{h_{dir}}{k_{tsv} \cdot A_{tsv}} + \frac{h_{dir}}{k_{Si} \cdot A_{Si}} \tag{7.2}$$

In the equation, dir represents the thermal dissipation direction (i.e., horizontal or vertical). h is the thickness of the material. k and A denote the thermal conductivity of the corresponding material per unit volume and the area of the corresponding material, respectively. When the lateral thermal conductivity of TSV is considered, k_{tsv} should be combined with TSV filling and liner materials.

7.1.3 3D Thermal Cycling Effect

Thermal cycling effect is another factor that can cause reliability issues of 3D ICs [188]. Fig. 7.1 shows the generated thermal expansion forces are highlighted by arrows, which is from the hotter blocks to the cooler blocks. As shown, the temperature in blocks 2, 4, and 6 are higher than that of blocks 1, 3, and 5. The corresponding forces labeled 1 and 3 are in the opposite direction with force 2. Forces in opposite directions cause cracks in the stacked chips and make the

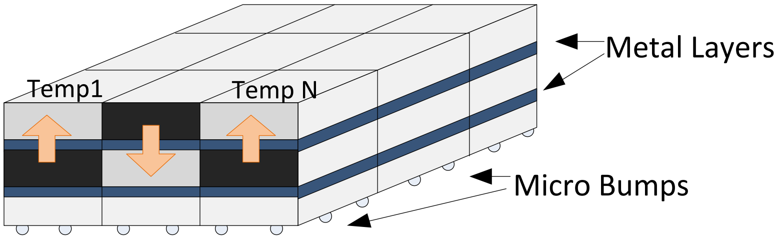

Figure 7.1. Stack level thermal cycling effect in 3D structure. Thermal stresses are pointing from hot blocks(dark color) to cool blocks(light color). Alternating direction of stresses (the arrows) easily cause cracking on thinned substrate.

thinned silicon substrate more vulnerable to be damaged. A run-time thermal cycling management scheme is proposed to eliminate the damaging thermal cycling pattern by using dynamic power scaling.

7.2 Thermomechanical Stress-Aware 3D Design Methodology

In this section, we present the detailed explanation of our proposed design-time thermal stress-aware floorplan technique and run-time thermal management scheme to achieve mechanical equilibrium thermal cycling pattern.

7.2.1 The Heuristic Floorplan Flow

The purpose of the design-time thermal stress-aware TSV management aims at reducing the CTE-induced thermomechanical stress. Based on Equation 7.1, the reduction of thermomechanical stress can be translated into the minimization of the TSVs thermal load.

Intuitively, placing TSV farms far away from the hot regions can reduce their thermal load. However, in most cases, the hotspots are the functional units that are most active and highly utilized, which usually indicates the requirement for the high connectivity. Timing/performance and other design goals prevent such thermal placement options. Moving TSV farms away from those hotspots induces larger wire length and communication delay. As a result, the thermal stress-aware floorplan is obligatory to sustain the high circuit performance without causing severe thermal reliability problems.

In addition to the traditional floorplan solutions that make great efforts on balancing the area and performance trade-off, the novel floorplan flow also strives to minimize the thermal-induced stresses at the same time. Circuit description and average power consumption of each block are given as inputs. The average power consumption of each block is estimated from the estimated power density on the chip. The circuit description consists of: 1) block descriptions, including the block name, area, and allowable aspect ratio (minimum and maximum aspect ratio during floorplanning); and 2) the connectivity information. TSV farms are treated as soft blocks in the floorplan with given thermal characteristics described in Section 7.1.

A simulated annealing based floorplanner is employed in the flow along with an analytical initial floorplan to speed up the convergence. The circuit is partitioned into required tiers by balancing the TSV number and chip area. Then, the initial floorplan is performed analytically by placing the modules that have low power density around TSV farm. Afterwards, the TSV and tier temperature are generated for initial cost calculation. After the initial floorplan, modules are randomly selected and permuted to obtain a better thermal distribution across the whole chip. Besides changing module position, the aspect ratio of modules can be adjusted. For TSV, changing the aspect ratio of TSV farm means adjusting the arrangement of fixed number TSVs. By adjusting the arrangement of TSVs, the lateral thermal path is changed for better heat dissipation.

A cost function (Equation 7.3) is developed to evaluate the floorplan from each simulated annealing iteration, where α, β, γ, and δ are associated weighting parameters. A is the final chip area, T_{avg} is the average temperature among blocks, T_{tsv} is the estimated average temperature of TSV farms, and W is the wire length in the design. A simple manhattan-distance based wire length model is used to estimated the wire length overhead.

$$Cost = \alpha * A + \beta * T_{avg} + \gamma * T_{tsv} + \delta * W \tag{7.3}$$

After each iteration, the cost is calculated based on the floorplan thermal profile to guide the floorplan. The iteration process terminates once the SA convergence condition is satisfied or maximum iteration step is reached.

7.2.2 Thermal Cycling-Aware Run-Time Management

Run-time thermal cycling management scheme is devised as the second stage of our thermal management methodology to achieve 3D architecture mechanical equilibrium by eliminating mechanical damaging cycling patterns as illustrated in Section 7.1.

The run-time management is performed following a bottom-up, layer-by-layer order for the whole stack. Each tier is partitioned in fine-granularity, and the temperature of each grid is monitored in runtime. Given the power trace derived from the supply voltage and activity factor of each block, a dynamic thermal profile can be obtained from temperature sensors in each sampling cycle. The proposed dynamic thermal management framework is shown in Figure 7.2. After the preliminary sampling period, the grid temperature on each tier is available and the temperature gradients can be captured[3].

The first step in the flow controls the temperature gradients of each grid to eliminate large temperature gradients between adjacent grids. Correspondingly, mechanical force vectors are generated based on this temperature gradients information. On the other hand, predefined thresholds are determined based on the TSV size, material, and substrate thickness. After the comparison between the force vectors and the predefined thresholds, if the force vectors are larger than thresholds, dynamic power scaling techniques, such as DVFS, will be deployed to control the thermal dissipation of hot grids.

In addition to the temperature gradients, the thermal cycling pattern should be handled carefully. After the control of temperature gradients, the cycling pattern is taken into account by comparing the force vectors of neighboring grids in two adjacent layers. If the thermal cycling pattern is in an alternating way as described in Figure 7.1, dynamic power management is applied to the high temperature regions to lower the resulting thermal mechanical stresses and achieve mechanical equilibrium. The power scaling results in new thermal cycling pattern in the stack, which produces further adjustment in next sampling interval till the whole stack

---

[3] In this work, the temperature differences between grids are used to represent temperature gradients for simplicity.

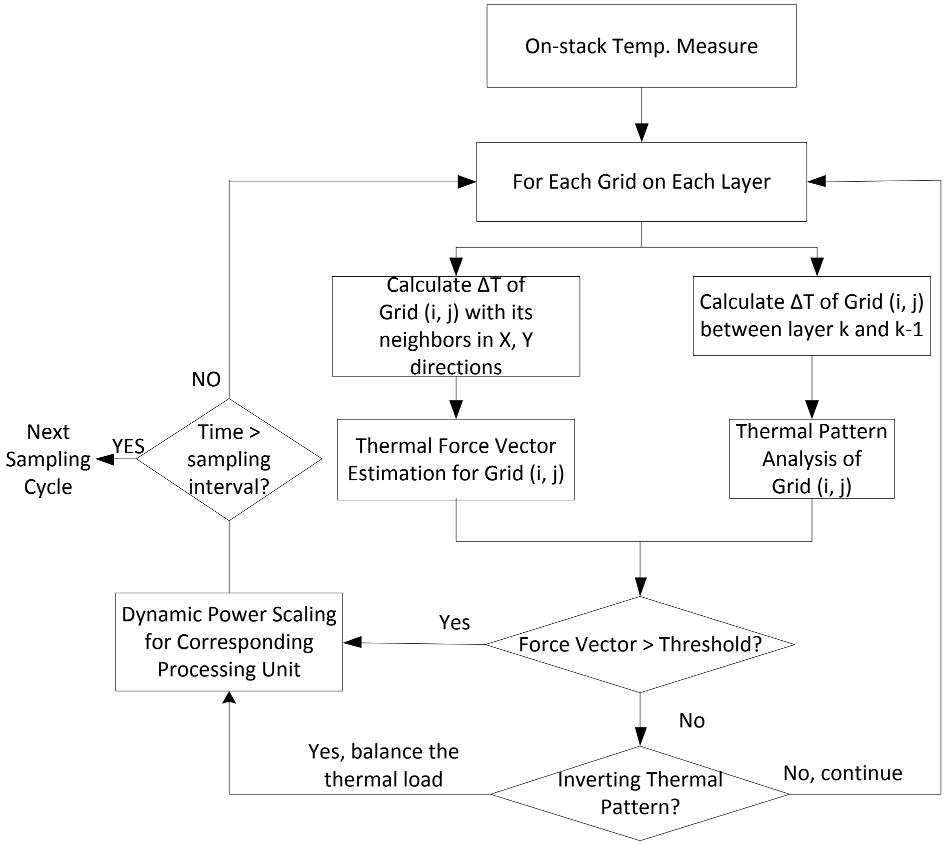
On-stack Temp. Measure
For Each Grid on Each Layer
Calculate ΔT of Grid (i, j) with its neighbors in X, Y directions
Calculate ΔT of Grid (i, j) between layer k and k-1
Thermal Force Vector Estimation for Grid (i, j)
Thermal Pattern Analysis of Grid (i, j)
NO
Time > sampling interval?
YES
Next Sampling Cycle
Dynamic Power Scaling for Corresponding Processing Unit
Yes
Force Vector > Threshold?
No
Yes, balance the thermal load
Inverting Thermal Pattern?
No, continue

Figure 7.2. Run-time thermal cycling-aware thermal management flow for one sampling cycle.

reaches the mechanical equilibrium state.

7.3 Experiment Results and Analysis

To evaluate the effect of the proposed thermal management methodology, a typical 3D floorplaner 3DFP [189] that aims at reducing the average on-chip temperature is employed as our baseline. We successfully implemented the thermal stress-aware floorplan mechanism in 3DFP. We also extend Hotspot [190] to handle 3D lateral and vertical thermal dissipation. The 3D Hotspot is used for circuit temperature evaluation after final floorplan. MCNC benchmarks and a core+memory 3D stacking system are leveraged in the block-level and system-level thermal stress-aware floorplan. We differentiate the block-level and system-level simulation since they have distinct thermal characteristics. The temperature gradients at block-level are

Table 7.1. Thermal parameters that are used in the experiments; sensitivity study is performed with various TSV lateral thermal conductivity and TSV diameter.

silicon thickness	100 μm
silicon thermal conductivity	120 $W/(mK)$
silicon heat capacity	1.75 $\times 10^6$ $J/(m^3K)$
thermal interface material (TIM) thickness	20 μm
TIM thermal conductivity	4 $W/(mK)$
TIM heat capacity	4 $\times 10^6$ $J/(m^3K)$
TSV lateral thermal conductivity	10 - 200 $W/(mK)$
Thermal Vias vertical thermal conductivity	350 $W/(mK)$
heat sink thermal conductivity	200 $W/(mK)$
TSV diameter	5 - 10 μm
Young's modulus	120GPa
Poisson's ration	0.30

larger and more unpredictable due to the various characteristics of the underlying circuits, while the temperature distribution at system-level is usually uniform and predictable because of the relatively regular placement of functional modules.

The thermal parameters that are used in the experiment are listed in Table 7.1. In additional to the mechanism evaluation, we vary the TSV lateral thermal conductivity and TSV diameter to study the parameter sensitivity. The TSV lateral thermal conductivity and diameter that we used for block- and system-level study are 170$W/(mK)$ [191] and 10μm, respectively. For simplicity, the elastic mismatch between silicon and copper is neglected.

7.3.1 Block-Level Thermomechanical Stress-Aware Floorplan

To quantify the TSV thermal load reduction from the stress-aware floorplan, we conduct experiments on five MCNC benchmarks: `ami33`, `ami49`, `hp`, `xerox`, `apte` and two industry benchmarks, marked as `industry1` and `industry2`. Similar to the approach in [189], these benchmarks are partitioned into two tiers at block level for simplicity. Note that higher level of partitioning can be supported with our framework. The characteristics of benchmarks vary in terms of block numbers, interconnect complexity and power density. These circuit information is extracted from the benchmark description files. The average power density for each circuit is within the rage of 0.5-2.4 W/mm^2. TSV farms are created based on the partitioning and interconnects information. During the simulation, all modules including

Table 7.2. Design time thermomechanical stress-aware floorplan results with average and peak on-chip temperature and thermal stresses reduction

Circuit	**Baseline**			**Proposed**			**Stress Reduction**
	avg T (K)	peak T (K)	TSV T (K)	avg T (K)	peak T (K)	TSV T (K)	(MPa)
ami33	349.2	389.2	357	346.4	376.4	346.4	25.44
ami49	329.45	374.12	317.48	329.83	355.2	314.25	7.752
apte	330.228	425.65	318.98	331.043	436	316.21	6.648
hp	325.74	351.5	337.67	325.56	362.4	313.35	58.37
xerox	321.97	339.21	320.7	321.93	339.03	313.94	16.224
industry1	314.8	317.7	313.4	312.4	315.2	312.5	2.16
industry2	313.31	319.14	307.19	310.2	318.1	306.38	1.944

Table 7.3. The normalized area/wire length and execution time comparison

Circuit	normalized area	normalized wire length	**Run Time (s) (proposed/baseline)**
ami33	1.047	0.914	16.69/43.89
ami49	0.98	1.007	201.79/198.56
apte	1.01	1.12	2.93/3.22
hp	1.02	1.28	5.58/11.5
xerox	0.996	0.926	6.81/22.34
industry1	1.128	0.866	702.2/1837
industry2	1.342	0.604	806.99/2968.18

TSV farms are treated as soft blocks with flexible aspect ratio. In addition, the weighting parameters for the average on-chip temperature and TSVs thermal load are the same in the cost function.

The experiment results are shown in Table 7.2. As shown in the table, the peak and average TSV temperature reduction are 28.29K and 7.63K, respectively. The peak temperatures have been reduced for most benchmarks (except *apte* and *hp*) after considering the TSV lateral thermal conduction. The TSV axial thermal stress reductions are listed in the last column, where the average thermal axial stress reduction is 16.934MPa. Table 7.3 reports the area and wirelength that are normalized to the baseline and the comparison of execution time between proposed scheme and thermal-aware scheme. The chip area for two circuits (*ami49* and *xerox*) are slightly reduced due to the re-shape of modules. The wirelengths are increased in benchmarks *ami49*, *apte*, and *hp* while decreased slightly in benchmarks *ami33*, *xerox*, and *industry1*. However, for benchmark *industry2*, the wirelength is dramatically reduced after TSV thermal aware floorplan.

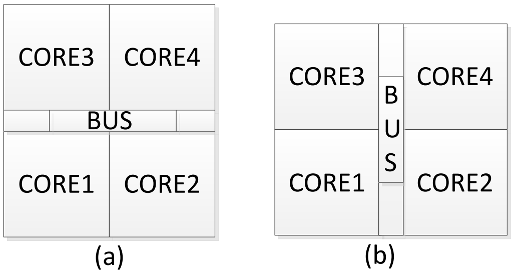

Figure 7.3. Floorplan of the core-layer before and after optimization. (a) TSV bus is placed horizontally in the floorplan for wire length and area driven floorplan; (b) TSV bus is placed vertically after thermomechanical stress aware floorplan.

7.3.2 System-Level Thermomechanical Stress-Aware Floorplan

In addition to the block-level simulation, we also applied the thermal management flow into a TSV-based 3D design for the system-level analysis. Different from block-level partitioning, the whole system has even higher connectivity requirement, indicating larger TSV occupancy in the floorplan. The 3D design stacks two tiers that have same size in a fashion of face-to-back bonding. The bottom layer mimics a multi-core processor, where four SPARC-like cores with private L2 cache are deployed. The top layer is assumed as a stacked memory chip that is divided into four blocks as the last level cache. TSV buses are integrated as inter-layer connection.

In the baseline, wire length and average temperature are the major metrics to guide the designs. In this way, TSV buses are placed in the middle of the chip to reduce routing length. Figure 7.3(a) illustrates the related baseline floorplan. The corresponding zoom-in temperature map is shown in Figure 7.4(a). It is obvious that TSV bus has high thermal load since it is surrounded around local hotspots. Due to the lateral thermal blockage effect of TSV bus [182], the heat dissipation path of local hotspots is blocked by TSVs, resulting in elevated temperature and steep temperature gradient.

In order to reduce the thermal load and thermal stresses of the TSV bus, the proposed thermal stresses management flow gives another solution after TSVs floorplan optimization as shown in Figure 7.3(b). The corresponding temperature

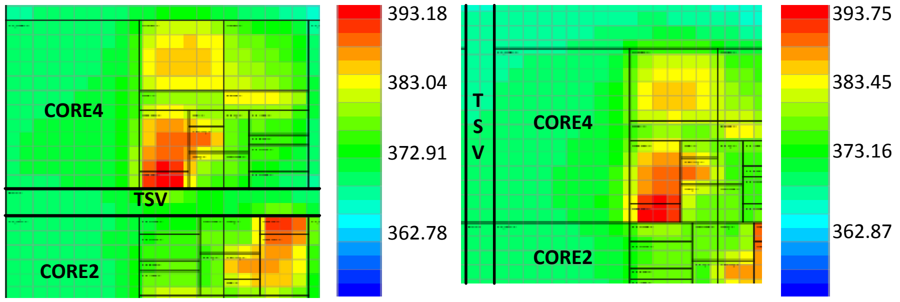

Figure 7.4. Zoomed in TSV thermal stress-aware floorplan on core-layer. (a) TSV bus is placed horizontally between execution units of two cores, resulting in higher TSV temperature; (b) TSV bus is placed vertically to for reduced thermal load.

map is shown in Figure 7.4(b). By placing the TSV bus away from local hotspots, the overall temperature on TSV bus is decreased. In general, the stress-aware floorplan can decreases the average temperature on TSV bus by 6.53K with 16.46MPa axial thermal stress reduction. The peak temperature of each core slightly increases about 4.38K because moving TSV bus away results in direct contact of local hotspots. The average temperature for upper layer memory is 369.54K. The relatively lower temperature on the top layer is beneficial from lower power density and uniform power distribution.

7.3.3 System-Level Run-time Thermal Management Scheme

The run-time thermal management scheme is examined on the optimized 3D chip in last section. In spite its time-consuming temperature evaluation of HotSpot, we leverage accurate thermal information provided by HotSpot in grid granularity for sensor simulation. Furthermore, the transient power trace of each block as well as the activity factor are also included to help evaluate run-time on-chip temperature.

As shown in Figure 7.5, the transient irregular temperature distribution on the memory layer can cause inverted thermal cycling pattern. Before power scaling, the inverted thermal cycling pattern may occur between the cooler region of lower core layer and the hotter region of upper memory layer. In order to have a better understanding of this thermal cycling effect, each memory block is further partitioned into three regions. The region in the memory (shown in Figure 7.5) causes the mechanical damaging thermal cycling pattern. After the power scaling on the corresponding memory region, the temperature of this region is decreased and the

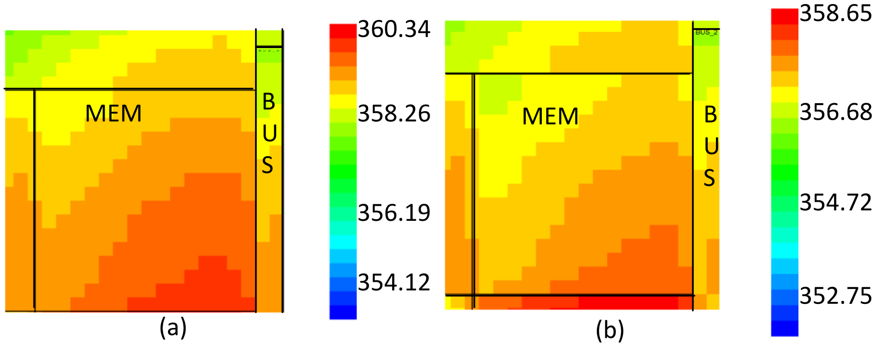

Figure 7.5. Zoomed in memory layer run-time thermal management results. The zoomed region causes inverting thermal cycling pattern between two layer. (a) The thermal map before power scaling; (b) thermal map after power scaling on interested region.

mechanical damaging thermal cycling pattern is eliminated. The average temperature on memory chip is reduced by 1.5K after the power scaling, resulting smaller temperature gradients between two layers. In this case study, the power scaling is performed by reducing the frequency from 1333MHz to 1066MHz on memory layer. The performance degradation is highly related to the application characteristics, on average, the performance degrades less than 5% after frequency scaling [192].

7.3.4 Sensitivity Study on TSV Thermal Conductivity and Diameter

With technology scaling, smaller TSVs and advanced materials are applied, therefore, the effect on various thermal conductivity should be studied. In this experiment set, we take benchmark `hp` as an example and show the variation of average on-chip temperature and TSV farm temperature. The TSV diameter is first fixed as $10\mu m$ and the thermal conductivity increases with the step of $20W/(mK)$. Then we fix the TSV lateral thermal conductivity as $170W/(mK)$ and increase the TSV diameter from $5\mu m$ to $10\mu m$ with incremental step as $1\mu m$. For each different setting, we first perform the TSV stress aware floorplan and then calculate the corresponding TSV temperature and average on-chip temperature. Figure 7.6 and 7.7 shows the temperature results of different TSV thermal conductivity and diameter settings.

From Figure 7.6, we can see that with the increased TSV thermal conductivity,

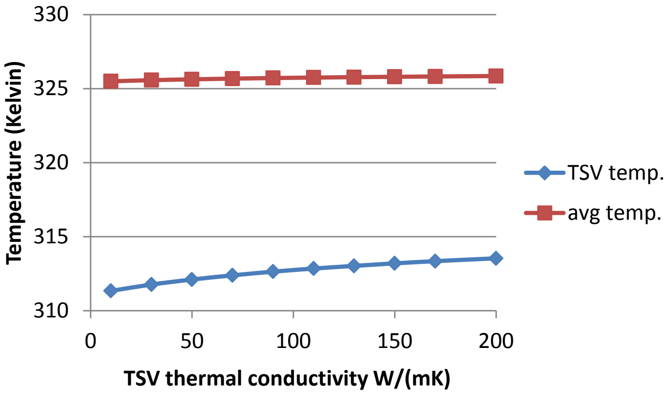

Figure 7.6. Temperatures with different TSV thermal conductivity settings.

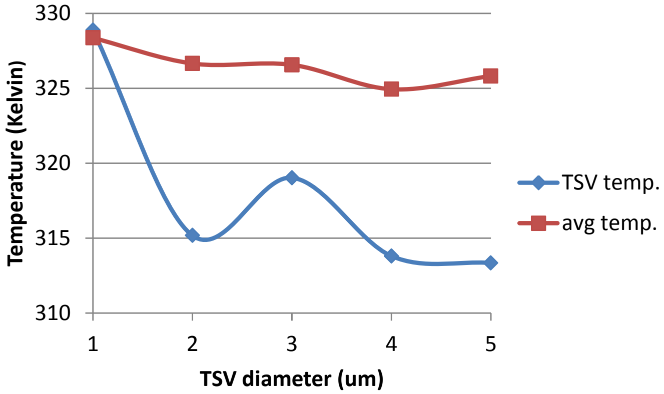

Figure 7.7. Temperature results with different TSV diameters.

the TSV temperature is actually increased. This result is counter-intuitive, however, it is reasonable since the TSV liner layer acts as a thermal shield between TSV and other modules. Once the thermal conductivity is increased, the TSV temperature is more susceptible to nearby modules. Nevertheless, the average on-chip temperature has negligible increment which is results from TSV temperature increases and nearby module temperature reductions (the lateral blockage effect is reduced with larger thermal conductivity).

Changing the TSV diameter results in smaller TSV farm area and larger TSV power consumption. Therefore, the power density is increased when we use smaller TSVs. In Figure 7.7, the results show that when we increase TSV diameter from $1\mu m$ to $5\mu m$, the temperature reduces from 329K to 313K. However, the temperature is not linearly reduced, when the TSV diameter is $3\mu m$, the temperature is larger than $2\mu m$ TSV because the power density is not linearly reduced. The

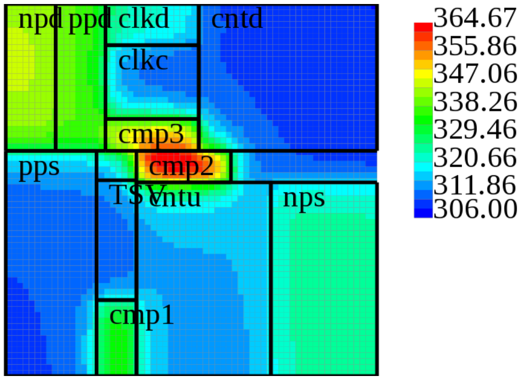

Figure 7.8. Thermal maps of benchmark *hp* without thermal vias.

average on-chip temperature is also reduced but with smaller range.

7.3.5 Impacts of Thermal Through-Silicon Vias

In order to handle the elevated temperature in 3D integrations, thermal vias are usually inserted to help building efficient vertical thermal dissipation path. Thermal vias are the vertical metal connections that only serve as heat removal without transmitting any signals [193]. To illustrate the effect of thermal vias and the impact of thermal via insertion on the floorplan, we use benchmark *hp* as an example. The TSV size is 5 μm and the lateral thermal conductivity is set as 170 $W/(mK)$. Other experiment configurations are consistent with values in Table 7.1. Since there is no signal transmission in thermal vias, we set the power consumption of thermal via as 0. For both scenarios with and without thermal vias, we perform the thermal stress-aware floorplan first and then the temperature estimation on each dedicated floorplan. The generated thermal maps are shown in Figure 7.8 and 7.9.

We can see from Figure 7.8, the component *cmp2* is the local hotspot due to the relatively larger power density. By placing the TSV between two cooler components, namely *ops* and *cntu*, the temperature on TSV farm can be lower than 320K. After inserting thermal vias, the peak temperature on chip drops from 364.67K to 359.06K as shown in Figure 7.9. By trying to reduce the average on chip temperature, the thermal vias are inserted near the hotspot *cmp2*. However, even though the TSV farm is moved away from the hotspot to reduce the TSV temperature, its position is constrained by the area and wire length. Therefore, in

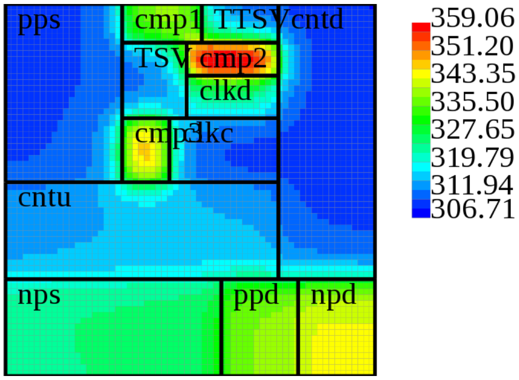

Figure 7.9. Thermal maps of benchmark *hp* with thermal vias.

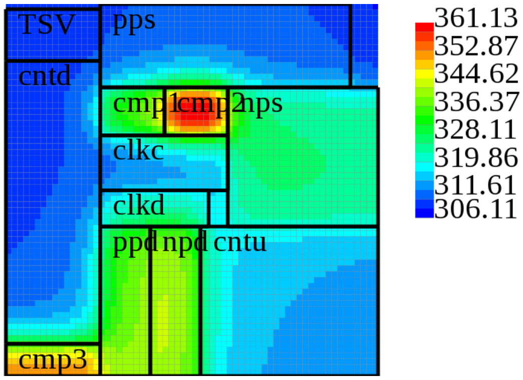

Figure 7.10. Thermal maps of benchmark *hp* when reducing the weight of wire length without thermal vias.

the next experiment, we decrease the weight of wire length by half and increase the weight of TSV temperature twice in the cost function during SA floorplanning. The thermal maps of designs with and without thermal vias are shown in Figure 7.10 and 7.11.

In the floorplan without thermal vias (Figure 7.10), the TSV farm moves far away from *cmp2* when the requirement on wire length is relax. The temperature on TSV farm reduce to less than $312K$. Moreover, the peak temperature is also reduced by $3.54K$ because the lateral thermal blockage effect of TSV farm has less impact on the hotspot. When adding thermal vias, the peak temperature reduces to $356.53K$. However, counterintuitively, thermal vias are not put directly near hotspot. This is because thermal vias only provide efficient vertical thermal dissipation; however, for horizontal direction, the dielectric layer still causes the lateral thermal blockage. Therefore, if thermal vias are near the hotspot on the

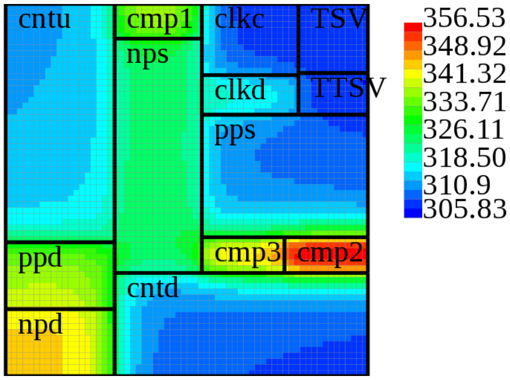

Figure 7.11. Thermal maps of benchmark *hp* when reducing the weight of wire length with thermal vias.

same tier, the temperature may be worse. Thermal vias can effectively help solving the thermal issue by making them close to the hotspot on different layers in the vertical direction.

7.4 Summary

This chapter presents a two-stage thermal management technique on design-time and run-time to alleviate the thermal challenges on 3D architectures. The design-time TSV thermal stress-aware floorplan technique aims at reducing the TSV thermal load during floorplan in design time. In the run time, thermal gradients and thermal cycling pattern induced mechanical reliability challenges are considered. Controlling the temperature gradients and eliminating damaging thermal cycling patterns can reduce the risk of cracking on the thinned silicon substrate. The results show that design-time floorplan can effectively reduce TSV thermal load for thermomechanical stresses minimization. The axial thermal stress reductions on average are 166.934MPa and 16.46MPa in block-level and system-level case studies, respectively. The temperatures with different thermal conductivities and TSV diameter are studied. In addition to signal TSVs, we also consider the floorplan with thermal vias. Experiment results illustrate that after run-time management, the core to memory stacking can achieve mechanical equilibrium on thermal cycling through dynamic power scaling with slightly performance overhead.

Chapter 8

Conclusion and Future Work

The increasing delay gap between transistors and interconnects makes the emerging 3D ICs appealing. With various benefits that are provided by 3D ICs, industries and academia are shifting design efforts from conventional 2D designs to 3D integrations. Several commercial demos and specifications (Xilinx FPGAs, HMCs, and HBMs) have proved the success of 3D designs. However, several challenges hinder the prevalent industrial adoption, and among these challenges, system cost and chip reliability are two primary concerns. This dissertation presents six studies managing the cost and reliability problems.

The first part of this dissertation introduces three cost-aware design methodologies. First, we observe that testing cost is one of the prominent factor in the system cost. Moreover, the difficulty in 3D testings prolongs the time-to-market, implicating higher hidden cost. There are several components in 3D testings: pre-bond tests, intermediate stacking tests, pre-package tests, and post-package tests. In general, the pre-bond and intermediate tests are used to improve the stacking yield for cost reduction. Nevertheless, if the chip yield is sufficient high, then these two tests can be eliminated to avoid extra testing cost and time. In this work, we build a cost analysis framework and explore the possibility of test elimination. One of benefits in 3D ICs is the potential of metal layer reduction. As shown in the second study, the mask cost is continuously increasing, especially, as the design becomes more and more complex, the requirement for metal layers is growing. In this study, the cost implication of explicit metal layer reduction is analyzed, that is, sacrificing the chip area for accommodating the routing with less metal layers.

Different from the second work that focusing on the chip level interconnects, the third work emphasizes the connections between routers. As higher bandwidth can be provided at the cost of larger number of TSVs (higher area and cost overhead), the third study explores the possibility of reusing those allocated but not utilized redundant TSVs. The study shows that performance improvement can be obtained with negligible overhead.

The second half of this dissertation presents three studies on 3D chip reliability problems. The first work examines the TSV lifetime under the influence of electromigration, especially the power network TSVs. The study reveals that the current distribution and correlation in TSV arrays have great impact on the final lifetime estimation. Without considering these factors in lifetime analysis may result in unexpected early chip failure. Next, due to the relatively large TSV size and the shorter distance between TSVs, the capacitive coupling between TSVs affects the signal integrity by introducing cross talk effects. The second study presents a coding method to alleviate the cross talk inferences with affordable power and area overhead. The third work considers the reliability problem induced from thermo-mechanical stresses. The TSV fabrication steps introduce thermal mechanical stresses which degrades the device performance and endangers the chip mechanical stability. Thus a two stage management scheme is proposed to tackle the stresses in both design-time and run-time.

Even though the studies developed in this dissertation solve the cost and reliability problems from various aspects, it is neither optimal nor complete. Several work can be done to further extent the cost saving designs. The test elimination work can be completed through gathering the practical manufacturing data from industry and differentiate different testing strategies. In the metal layer reduction study, the detailed interconnect modeling is not considered. For example, the different electrical characteristics and aspect ratio of local, semi-global, and global interconnects should be modeled and the corresponding optimal and maximum wirelength can be calculated to estimate the optimal number of metal layers. Alternatively, by applying explicit metal layer reduction, the performance impact can be another attractive topic. Increasing the chip area to accommodate more wires per tier can reduce the required number of metal layers, however, the underlying circuit placement should be updated to guarantee the timing constraints. Moreover, the increased routing area given that the elimination of long global wires reduces the number of repeaters also implies more routing resource per layers. A detailed TSV failure modeling can capture the performance improvement of the proposed reconfigurable 3D NoC more precisely, such as TSV failure model with transmission loads or manufacturing limitations.

More research topics are remained to be explored in the 3D integration era, especially when the transistor scales deep into the sub-micron region. At the system design level, heterogeneous stacking can be a potential design option for both better performance and low cost. There are two approaches for heterogeneous integration: integration of different functionalities or integration of different technology generations. The first kind of integrations include mixed-signal circuit designs, CPU and GPU or accelerators co-processors, CMOS compatible and incompatible devices integrations. The second kind of heterogeneous stacking enables utilizing devices with different technology nodes. The devices in the advanced technology generation have higher switching speed and more compact size. However, the leakage power becomes dominant. Therefore, we can build the latency sensitive circuits with advanced technology while the power sensitive components with low cost mature technologies. The impact of heterogeneous stacking leveraging 3D integrations on cost and performance is left to be explored. In addition to the cost, numerous reliability related topics are remained as future work, such as the reliable clock tree and power network design. For example, the clock tree design should be robust and skew balanced even under sporadical TSV failures. Since TSV redundancy is the prevalent strategy towards TSV failures, the clock signal shifting from one TSV to its redundant may cause the clock skew and slew problems. Therefore, at the design time, the suitable alternative clock path should be considered in advance. In addition to the clock network, the power network design should consider reducing the IR droop and avoiding the temperature influences from self heating. As the number of layers in 3D stacking grows, the power delivery becomes more challenging as the distance between power source and top-most die is increased. Under such circumstance, the allocation and placement of power TSVs are critical, especially for power-hungry circuits.

Bibliography

[1] Moore, G. E. (1965) "Cramming more components onto integrated circuits," *Electronics Magazine*, **86**(1), pp. 82–85.

[2] Banerjee, K., S. Souri, P. Kapur, and K. Saraswat (2001) "3-D ICs: a novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration," *Proceedings of the IEEE*, **89**(5), pp. 602–633.

[3] Meindl, J. (2003) "Interconnect opportunities for gigascale integration," *IEEE Micro*, **23**(3), pp. 28–35.

[4] Xie, Y., G. H. Loh, B. Black, and K. Bernstein (2006) "Design space exploration for 3D architectures," *Journal of Emerging Technologies in Computing Systems*, **2**(2), pp. 65–103.

[5] Davis, W. R., J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. Franzon (2005) "Demystifying 3D ICs: the pros and cons of going vertical," *IEEE Design Test of Comoputer*, **22**(6), pp. 498–510.

[6] Knickerbocker, J. U., P. S. Andry, B. Dang, R. R. Horton, M. J. Interrante, C. S. Patel, R. J. Polastre, K. Sakuma, R. Sirdeshmukh, E. J. Sprogis, S. M. Sri-Jayantha, A. M. Stephen, A. W. Topol, C. K. Tsang, B. C. Webb, and S. L. Wright (2008) "Three-dimensional silicon integration," *IBM Journal of Research and Development*, **52**(6), pp. 553–569.

[7] Ghosal, P. and S. Chatterjee (2012) "Partitioning in 3D ICs: a TSV aware strategy with area balancing," in *Proceedings of International Conference on Devices, Circuits and Systems.*

[8] Chang, H.-L., H.-C. Lai, T.-Y. Hsueh, W.-K. Cheng, and M. C. Chi (2012) "A 3D IC designs partitioning algorithm with power consideration," in *Proceedings of International Symposium on Quality Electronic Design.*

[9] Hentschke, R., S. Sawicki, M. Johann, and R. Reis (2006) "An algorithm for I/O partitioning targeting 3D circuits and tts impact on 3D-vias," in *Proceedings of International Conference on Very Large Scale Integration.*

[10] Panth, S., K. Samadi, Y. Du, and S. Lim (2015) "PD14 placement-driven partitioning for congestion mitigation in monolithic 3D IC designs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **pp**(99), pp. 1–1.

[11] Jiang, I.-R. (2009) "Generic integer linear programming formulation for 3D IC partitioning," in *Proceedings of International SOC Conference.*

[12] Kumar, A., S. Reddy, B. Becker, and I. Pomeranz (2012) "Performance aware partitioning for 3D-SOCs," in *Proceedings of International SoC Design Conference.*

[13] Banerjee, S. and S. Majumder (2014) "A thermal aware 3D IC partitioning technique," in *Proceedings of International Symposium on VLSI Design and Test.*

[14] Loh, G. H., Y. Xie, and B. Black (2007) "Processor design in 3D die-stacking technologies," *IEEE Micro*, **27**(3), pp. 31–48.

[15] Khan, A., R. Vatsa, S. Roy, and B. Das (2014) "A new efficient topological structure for floorplanning in 3D VLSI physical design," in *Proceedings of International Advance Computing Conference.*

[16] Hayashi, R., H. Ohta, and K. Fujiyoshi (2012) "A novel representation for 3D-LSI floorplan: merged FT Squeeze," in *Proceedings of Latin American Symposium on Circuits and Systems.*

[17] Cong, J., J. Wei, and Y. Zhang (2004) "A thermal-driven floorplanning algorithm for 3D ICs," in *Proceedings of International Conference on Computer Aided Design.*

[18] Tsai, M.-C., T.-C. Wang, and T. Hwang (2011) "Through-silicon via planning in 3-D floorplanning," *IEEE Transactions on Very Large Scale Integration Systems*, **19**(8), pp. 1448–1457.

[19] Jain, A., S. Alam, S. Pozder, and R. Jones (2011) "Thermal-electrical co-optimisation of floorplanning of three-dimensional integrated circuits under manufacturing and physical design constraints," *IET Computers Digital Techniques*, **5**(3), pp. 169–178.

[20] Li, W., J. Kim, and J.-W. Chong (2012) "A novel congestion estimation model and congestion aware floorplan for 3D ICs," in *Proceedings of International Conference on Innovation Management and Technology Research.*

[21] Tezuka, H. and K. Fujiyoshi (2012) "An efficient solution space for floorplan of 3D-LSI," in *Proceedings of International Conference on Electronics, Circuits and Systems.*

[22] Li, X., Y. Ma, and X. Hong (2009) "A novel thermal optimization flow using incremental floorplanning for 3D ICs," in *Proceedings of Asia and South Pacific Design Automation Conference.*

[23] Jung, M., T. Song, Y. Wan, Y. Peng, and S. K. Lim (2014) "On enhancing power benefits in 3D ICs: block folding and bonding styles perspective," in *Proceedings of Design Automation Conference*, pp. 4:1–4:6.

[24] Falkenstern, P., Y. Xie, Y.-W. Chang, and Y. Wang (2010) "Three-dimensional integrated circuits (3D ICs) floorplan and power/ground network co-synthesis," in *Proceedings of Asia and South Pacific Design Automation Conference.*

[25] Hsu, M.-K., Y.-W. Chang, and V. Balabanov (2011) "TSV-aware analytical placement for 3D IC designs," in *Proceedings of Design Automation Conference.*

[26] Thorolfsson, T., G. Luo, J. Cong, and P. Franzon (2010) "Logic-on-logic 3D integration and placement," in *Proceedings of International 3D Systems Integration Conference.*

[27] Yan, H., Q. Zhou, and X. Hong (2008) "Efficient thermal aware placement approach integrated with 3D DCT placement algorithm," in *International Symposium on Quality Electronic Design.*

[28] Zhuang, H., J. Lu, K. Samadi, Y. Du, and C.-K. Cheng (2013) "Performance-driven placement for design of rotation and right arithmetic shifters in monolithic 3D ICs," in *Proceedings of International Conference on Communications, Circuits and Systems.*

[29] Ababei, C., Y. Feng, B. Goplen, H. Mogal, T. Zhang, K. Bazargan, and S. Sapatnekar (2005) "Placement and routing in 3D integrated circuits," *IEEE Design Test of Computers*, **22**(6), pp. 520–531.

[30] Cong, J., G. Luo, J. Wei, and Y. Zhang (2007) "Thermal-aware 3D IC placement via transformation," in *Proceedings of Asia and South Pacific Design Automation Conference.*

[31] Athikulwongse, K., M. Ekpanyapong, and S. K. Lim (2014) "Exploiting Die-to-Die Thermal Coupling in 3-D IC Placement," *IEEE Transactions on Very Large Scale Integration Systems,*, **22**(10), pp. 2145–2155.

[32] Cong, J., G. Luo, and Y. Shi (2011) "Thermal-aware cell and through-silicon-via co-placement for 3D ICs," in *Proceedings of Design Automation Conference*.

[33] Cong, J. and Y. Zhang (2005) "Thermal-driven multilevel routing for 3D ICs," in *Proceedings of Asia and South Pacific Design Automation Conference*.

[34] Zhang, T., Y. Zhan, and S. Sapatnekar (2006) "Temperature-aware routing in 3D ICs," in *Asia and South Pacific Conference on Design Automation*.

[35] Pak, J., S. K. Lim, and D. Pan (2012) "Electromigration-aware routing for 3D ICs with stress-aware EM modeling," in *Proceedings of International Conference on Computer-Aided Design*.

[36] Cheng, L., W. Hung, G. Yang, and X. Song (2004) "Congestion estimation for 3D routing," in *Proceedings of Computer society Annual Symposium on VLSI*.

[37] Roy, D., P. Ghosal, and S. Mohanty (2014) "FuzzRoute: A Method for Thermally Efficient Congestion Free Global Routing in 3D ICs," in *Proceedings of Computer Society Annual Symposium on VLSI*.

[38] Hsu, P.-Y., H.-T. Chen, and T. Hwang (2013) "Stacking signal TSV for thermal dissipation in global routing for 3D IC," in *Proceedings of Asia and South Pacific Design Automation Conference*.

[39] Pathak, M. and S. K. Lim (2009) "Performance and Thermal-Aware Steiner Routing for 3-D Stacked ICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **28**(9), pp. 1373–1386.

[40] Chang, C.-J., P.-J. Huang, T.-C. Chen, and C.-N. Liu (2011) "ILP-based inter-die routing for 3D ICs," in *Proceedings of Asia and South Pacific Design Automation Conference*.

[41] Emma, P. and E. Kursun (2010) "3D system design: A case for building customized modular systems in 3D," in *Proceedings of International Interconnect Technology Conference*.

[42] Diamantopoulos, D., K. Siozios, D. Bekiaris, and D. Soudris (2011) "A novel methodology for architecture-level exploration of 3D SoCs," in *Proceedings of International Conference on Design Technology of Integrated Systems in Nanoscale Era.*

[43] Milojevic, D., T. Carlson, K. Croes, R. Radojcic, D. Ragett, D. Seynhaeve, F. Angiolini, G. Van der Plas, and P. Marchal (2009) "Automated Pathfinding tool chain for 3D-stacked integrated circuits: practical case study," in *Proceedings of International Conference on 3D System Integration.*

[44] Krishnan, V. and S. Katkoori (2007) "A 3D-layout aware binding algorithm for high-level synthesis of three-dimensional integrated circuits," in *Proceedings of International Symposium on Quality Electronic Design.*

[45] Chen, Y., G. Sun, Q. Zou, and Y. Xie (2012) "3DHLS: Incorporating high-level synthesis in physical planning of three-dimensional (3D) ICs," in *Proceedings of Design, Automation Test in Europe Conference Exhibition.*

[46] Zou, Q., Y. Chen, Y. Xie, and A. Su (2011) "System-level design space exploration for three-dimensional (3D) SoCs," in *Proceedings of International Conference on Hardware/Software Codesign and System Synthesis.*

[47] Lin, C.-H., W.-T. Hsieh, H.-C. Hsieh, C.-N. Liu, and J.-C. Yeh (2011) "System-level design exploration for 3-D stacked memory architectures," in *Proceedings of International Conference on Hardware/Software Codesign and System Synthesis.*

[48] Priyadarshi, S., J. Hu, W. H. Choi, S. Melamed, X. Chen, W. Davis, and P. Franzon (2012) "Pathfinder 3D: A flow for system-level design space exploration," in *Proceedings of International 3D Systems Integration Conference.*

[49] Jabbar, M., D. Houzet, and O. Hammami (2013) "Heterogeneous stacking of 3D MPSoC architecture: Physical implementation analysis and performance evaluation," in *Proceedings of Asia Symposium on Quality Electronic Design.*

[50] Zhang, T., A. Cevrero, G. Beanato, P. Athanasopoulos, A. K. Coskun, and Y. Leblebici (2013) "3D-MMC: A modular 3D multi-core architecture with efficient resource pooling," in *Proceedings of Design, Automation Test in Europe Conference Exhibition.*

[51] Zhao, Q., Y. Iwai, M. Amagasaki, M. Iida, and T. Sueyoshi (2011) "A novel reconfigurable logic device base on 3D stack technology," in *Proceedings of International 3D Systems Integration Conference.*

[52] Loh, G. (2008) "3D-Stacked Memory Architectures for Multi-core Processors," in *Proceedings of International Symposium on Computer Architecture*.

[53] Woo, D. H., N. H. Seong, D. Lewis, and H.-H. Lee (2010) "An optimized 3D-stacked memory architecture by exploiting excessive, high-density TSV bandwidth," in *Proceedings of International Symposium on High Performance Computer Architecture*.

[54] Sim, J., A. Alameldeen, Z. Chishti, C. Wilkerson, and H. Kim (2014) "Transparent Hardware Management of Stacked DRAM as Part of Memory," in *Proceedings of International Symposium on Microarchitecture*.

[55] Zhu, Q., B. Akin, H. Sumbul, F. Sadi, J. Hoe, L. Pileggi, and F. Franchetti (2013) "A 3D-stacked logic-in-memory accelerator for application-specific data intensive computing," in *Proceedings of International 3D Systems Integration Conference*.

[56] Huang, C.-M., C.-C. Yang, C.-M. Wu, C.-C. Chiu, Y.-J. Liu, C.-C. Chu, C. Nien-Hsiang, W.-C. Chen, C.-H. Lin, and H.-H. Luo (2012) "A novel design flow for a 3D heterogeneous system prototyping platform," in *Proceedings of International SOC Conference*.

[57] Miura, N., Y. Koizumi, Y. Take, H. Matsutani, T. Kuroda, H. Amano, R. Sakamoto, M. Namiki, K. Usami, M. Kondo, and H. Nakamura (2013) "A Scalable 3D Heterogeneous Multicore with an Inductive ThruChip Interface," *IEEE Micro*, **33**(6), pp. 6–15.

[58] Liu, W., Y. Wang, X. Feng, Y. Huang, H. Yang, Y. Xie, and G. Chen (2014) "Exploration of electrical and novel optical chip-to-chip interconnects," *IEEE Design Test*, **31**(5), pp. 28–35.

[59] Bakir, M., C. King, D. Sekar, and B. Dang (2008) "Electrical, optical, and fluidic interconnect networks for 3D heterogeneous integrated systems," in *Proceedings of Avionics, Fiber-Optics and Photonics Technology Conference*.

[60] Lee, K.-W., A. Noriki, K. Kiyoyama, S. Kanno, R. Kobayashi, W.-C. Jeong, J.-C. Bea, T. Fukushima, T. Tanaka, and M. Koyanagi (2009) "3D heterogeneous opto-electronic integration technology for system-on-silicon (SOS)," in *International Electron Devices Meeting*.

[61] Sun, G., X. Dong, Y. Xie, J. Li, and Y. Chen (2009) "A novel architecture of the 3D stacked MRAM L2 cache for CMPs," in *Proceedings of International Symposium on High Performance Computer Architecture*.

[62] Mishra, A., X. Dong, G. Sun, Y. Xie, N. Vijaykrishnan, and C. Das (2011) "Architecting on-chip interconnects for stacked 3D STT-RAM caches in CMPs," in *Proceedings of International Symposium on Computer Architecture*.

[63] Wang, Y., C. Zhang, R. Nadipalli, H. Yu, and R. Weerasekera (2012) "Design exploration of 3D stacked non-volatile memory by conductive bridge based crossbar," in *Proceedings of International 3D Systems Integration Conference*.

[64] Koyanagi, M., K.-W. Lee, T. Fukushima, and T. Tanaka (2012) "Heterogeneous 3D integration technology and new 3D LSIs," in *Proceedings of International Conference on Solid-State and Integrated Circuit Technology*.

[65] Gadfort, P., A. Dasu, A. Akoglu, Y. K. Leow, and M. Fritze (2014) "A power efficient reconfigurable system-in-stack: 3D integration of accelerators, FPGAs, and DRAM," in *Proceedings of International System-on-Chip Conference*.

[66] Belhadj, B., A. Valentian, P. Vivet, M. Duranton, L. He, and O. Temam (2014) "The improbable but highly appropriate marriage of 3D stacking and neuromorphic accelerators," in *Proceedings of International Conference on Compilers, Architecture and Synthesis for Embedded Systems*.

[67] Samoilov, A., K. Tran, N. Kerness, J. Jones, P. McNally, S. Barnett, T. Parent, J. Ellul, A. Srivastava, K. Ikeuchi, T. Wang, and T. Zhou (2013) "3D heterogeneous integration for analog," in *Proceedings of International Electron Devices Meeting*.

[68] Liu, W., G. Chen, X. Han, Y. Wang, Y. Xie, and H. Yang (2014) "Design methodologies for 3D mixed signal integrated circuits: A practical 12-bit SAR ADC design case," in *Proceedings of Design Automation Conference*.

[69] Javdjic, D., G. H. Loh, C. Kaynak, and B. Falsafi (2014) "Unison cache: a scalable and effective die-stacked DRAM cache," in *Proceedings of International Symposium on Microarchitecture*.

[70] Pugsley, S., J. Jestes, R. Balasubramonian, V. Srinivasan, A. Buyuktosunoglu, A. Davis, and F. Li (2014) "Comparing implementations of near-data computing with in-memory MapReduce workloads," *IEEE Micro*, **34**(4), pp. 44–52.

[71] "JEDEC High Bandwidth Memory," .
URL `http://www.jedec.org/category/technology-focus-area/3d-ics-0`

[72] "Micron Hybrid Memory Cube," .
URL `http://www.hybridmemorycube.org/`

[73] "Xilinx Virtex 7," .
URL `http://www.xilinx.com/products/silicon-devices/3dic.html`

[74] Kearney, D., T. Hilt, and P. Pham (2012) "A liquid cooling solution for temperature redistribution in 3D IC architectures," *Microelectronics Journal*, **43**(9), pp. 602–610.

[75] Mizunuma, H., C.-L. Yang, and Y.-C. Lu (2009) "Thermal modeling for 3D-ICs with integrated microchannel cooling," in *Proceedings of International Conference on Computer-Aided Design*.

[76] Sekar, D., C. King, B. Dang, T. Spencer, H. Thacker, P. Joseph, M. Bakir, and J. Meindl (2008) "A 3D-IC technology with integrated microchannel cooling," in *Proceedings of International Interconnect Technology Conference*.

[77] Wen, C.-C., Y.-J. Chen, and S.-J. Ruan (2013) "Cluster-based thermal-aware 3D-floorplanning technique with post-floorplan TTSV insertion at via-channels," in *Proceedings of Asia Symposium on Quality Electronic Design*, pp. 200–207.

[78] Chakrabarty, K., S. Deutsch, H. Thapliyal, and F. Ye (2012) "TSV defects and TSV-induced circuit failures: The third dimension in test and design-for-test," in *Proceedings of International Reliability Physics Symposium*.

[79] Lwo, B.-J., Frank, M.-S. Lin, and K.-H. Huang (2014) "TSV reliability model under various stress tests," in *Proceedings of Electronic Components and Technology Conference*.

[80] Choi, H.-J., S.-M. Choi, M.-S. Yeo, S.-D. Cho, D.-C. Baek, and J. Park (2012) "An experimental study on the TSV reliability: Electromigration (EM) and time dependant dielectric breakdown (TDDB)," in *Proceedings of International Interconnect Technology Conference*.

[81] Okoro, C., J. Lau, F. Golshany, K. Hummler, and Y. Obeng (2014) "A detailed failure analysis examination of the effect of thermal cycling on Cu TSV reliability," *IEEE Transactions on Electron Devices*, **61**(1), pp. 15–22.

[82] Croes, K., V. Cherman, Y. Li, L. Zhao, Y. Barbarin, J. De Messemaeker, Y. Civale, D. Velenis, M. Stucchi, T. Kauerauf, A. Redolfi, B. Dimcic, A. Ivankovic, G. Van der Plas, I. De Wolf,

G. Beyer, B. Swinnen, Z. Tokei, and E. Beyne (2012) “Reliability concerns in copper TSV’s: Methods and results,” in *Proceedings of International Symposium on the Physical and Failure Analysis of Integrated Circuits*.

[83] Lee, B., B. Ahn, J. Kim, M. Kim, and J. Chong (2012) “A novel methodology for power delivery network optimization in 3-D ICs using through-silicon-via technology,” in *Proceedings of International Symposium on Circuits and Systems*.

[84] Corbalan, M., A. Keval, T. Toms, D. Lisk, R. Radojcic, and M. Nowak (2013) “Power and signal integrity challenges in 3D systems,” in *Proceedings of Design Automation Conference*.

[85] Song, T., C. Liu, Y. Peng, and S. K. Lim (2013) “Full-chip multiple TSV-to-TSV coupling extraction and optimization in 3D ICs,” in *Proceedings of the 50th Annual Design Automation Conference*.

[86] Chen, H.-T., H.-L. Lin, Z.-C. Wang, and T. Hwang (2011) “A new architecture for power network in 3D IC,” in *Proceedings of Design Automation and Test in Europe*.

[87] Pavlidis, V., I. Savidis, and E. Friedman (2011) “Clock distribution networks in 3-D integrated systems,” *IEEE Transactions on Very Large Scale Integration Systems*, **19**(12), pp. 2256–2266.

[88] Cho, J. and J. Kim (2013) “Signal integrity design of TSV and interposer in 3D-IC,” in *Proceedings of Latin American Symposium on Circuits and Systems*.

[89] Sung, H., K. Cho, K. Yoon, and S. Kang (2014) “A delay test architecture for TSV with resistive open defects in 3-D stacked memories,” *IEEE Transactions on Very Large Scale Integration Systems*, **22**(11), pp. 2380–2387.

[90] Xu, Q., L. Jiang, H. Li, and B. Eklow (2012) “Yield enhancement for 3D-stacked ICs: recent advances and challenges,” in *Proceedings of Asia and South Pacific Design Automation Conference*.

[91] Cheng, Y., L. Zhang, Y. Han, J. Liu, and X. Li (2011) “Wrapper chain design for testing tsvs minimization in circuit-partitioned 3D SoC,” in *Proceedings of Asian Test Symposium*.

[92] Lewis, D. and H.-H. Lee (2009) “Testing circuit-partitioned 3D IC designs,” in *Proceedings of Computer Society Annual Symposium on VLSI*.

[93] Millican, S. and K. Saluja (2014) "A test partitioning technique for scheduling tests for thermally constrained 3D integrated circuits," in *Proceedings of International Conference on VLSI Design and Embedded Systems*.

[94] Agrawal, M. and K. Chakrabarty (2013) "Test-cost optimization and test-flow selection for 3D-stacked ICs," in *Proceedings of VLSI Test Symposium*, vol. 0, IEEE Computer Society, Los Alamitos, CA, USA, pp. 1–6.

[95] Zhao, J., X. Dong, and Y. Xie (2010) "Cost-aware three-dimensional (3D) many-core multiprocessor design," in *Proceedings of Design Automation Conference*.

[96] Dong, X., J. Zhao, and Y. Xie (2010) "Fabrication cost analysis and cost-aware design space exploration for 3-D ICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **29**(12), pp. 1959–1972.

[97] Chen, Y., D. Niu, Y. Xie, and K. Chakrabarty (2010) "Cost-effective integration of three-dimensional (3D) ICs emphasizing testing cost analysis," in *Proceedings of International Conference on Computer-Aided Design*.

[98] Taouil, M., S. Hamdioui, and E. Marinissen (2011) "How significant will be the test cost share for 3D die-to-wafer stacked-ICs?" in *Proceedings of International Conference on Design Technology of Integrated Systems in Nanoscale Era*.

[99] Lee, H.-H. and K. Chakrabarty (2009) "Test challenges for 3D integrated circuits," *IEEE Design Test of Computers*, **26**(99), pp. 26–35.

[100] Noia, B. and K. Chakrabarty (2011) "Pre-bond probing of TSVs in 3D stacked ICs," in *Proceedings of International Test Conference*.

[101] Hamdioui, S. and M. Taouil (2011) "Yield improvement and test cost optimization for 3D stacked ICs," in *Proceedings of Asian Test Symposium*.

[102] Mercier, P., S. R. Singh, K. Iniewski, B. Moore, and P. O'Shea (2006) "Yield and cost modeling for 3D chip stack technologies," in *Proceedings of Custom Integrated Circuits Conference*.

[103] Pukite, P. and C. Berman (1990) "Defect cluster analysis for wafer-scale integration," *IEEE Transactions on Semiconductor Manufacturing*, **3**(3), pp. 128–135.

[104] L.-I. Tong, D.-L. C., C.-H. Wang (2007) "Development of a new cluster index for wafer defects," *The International Journal of Advanced Manufacturing Technology*, **31**(1), pp. 705–715.

[105] Zou, Q., J. Xie, and Y. Xie (2013) "Cost-driven 3D design optimization with metal layer reduction technique," in *Proceedings of International Symposium on Quality Electronic Design*.

[106] Taouil, M., S. Hamdioui, J. Verbree, and E. Marinissen (2010) "On maximizing the compound yield for 3D Wafer-to-Wafer stacked ICs," in *Proceedings of International Test Conference*.

[107] Verbree, J., E. Marinissen, P. Roussel, and D. Velenis (2010) "On the cost-effectiveness of matching repositories of pre-tested wafers for wafer-to-wafer 3D chip stacking," in *Proceedings of European Test Symposium*.

[108] Ferri, C., S. Reda, and R. I. Bahar (2008) "Parametric yield management for 3D ICs: Models and strategies for improvement," *ACM Journal on Emerging Technologies in Computing Systems*, **4**(4), pp. 1–22.

[109] Jiang, L., R. Ye, and Q. Xu (2010) "Yield enhancement for 3D-stacked memory by redundancy sharing across dies," in *Proceedings of International Conference on Computer-Aided Design*.

[110] Jiang, L., Q. Xu, and B. Eklow (2012) "On effective TSV repair for 3D-stacked ICs," in *Proceedings of Design, Automation Test in Europe Conference Exhibition*.

[111] Friedman, D., M. Hansen, V. Nair, and D. James (1997) "Model-free estimation of defect clustering in integrated circuit fabrication," *IEEE Transactions on Semiconductor Manufacturing*, **10**(3), pp. 344–359.

[112] Li, S., J.-H. Ahn, R. Strong, J. Brockman, D. Tullsen, and N. Jouppi (2009) "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Proceedings of International Symposium on Microarchitecture*.

[113] Weerasekera, R., L.-R. Zheng, D. Pamunuwa, and H. Tenhunen (2007) "Extending systems-on-chip to the third dimension: performance, cost and technological tradeoffs," in *Proceedings of International Conference on Computer-Aided Design*, pp. 212 –219.

[114] Davis, J., V. De, and J. Meindl (1998) "A stochastic wire-length distribution for gigascale integration (GSI). I. Derivation and validation," *IEEE Transactions on Electron Devices*, **45**(3), pp. 580–589.

[115] Landman, B. and R. Russo (1971) "On a pin versus block relationship for partitions of logic graphs," *IEEE Transactions on Computers*, **C-20**(12), pp. 1469–1479.

[116] Christie, P. (2000) "Rent exponent prediction methods," *IEEE Transactions on Very Large Scale Integration Systems,*, **8**(6), pp. 679–688.

[117] Kahng, A., S. Mantik, and D. Stroobandt (2001) "Toward accurate models of achievable routing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **20**(5), pp. 648–659.

[118] "NanGate FreePDK45 Generic Open Cell Library," .
URL `http://www.si2.org/openeda.si2.org/projects/nangatelib`

[119] "IC Cost Model, 2012 revision 1202," .

[120] Dally, W. J. and B. Towles (2001) "Route packets, not wires: on-chip inteconnection networks," in *Proceedings of Design Automation Conference*.

[121] Xie, Y., N. Vijaykrishnan, and C. Das (2010) "Three-dimensional network-on-chip architecture," in *Three-Dimensional Integrated Circuit Design: EDA, Design and Microarchitectures*, Springer.

[122] Feero, B. and P. Pande (2007) "Performance evaluation for three-dimensional networks-on-chip," in *Proceedings of Computer Society Annual Symposium on VLSI*.

[123] Li, F., C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir (2006) "Design and management of 3D chip multiprocessors using network-in-memory," *Computer Architecture News*, **34**(2), pp. 130–141.

[124] Lan, Y.-C., S.-H. Lo, Y.-C. Lin, Y.-H. Hu, and S.-J. Chen (2009) "BiNoC: A Bidirectional NoC Architecture with Dynamic Self-reconfigurable Channel," in *Proceedings of International Symposium on Networks-on-Chip*.

[125] Hesse, R., J. Nicholls, and N. Jerger (2012) "Fine-grained bandwidth adaptivity in networks-on-chip using bidirectional channels," in *Proceedings of International Symposium on Networks on Chip*.

[126] Xu, T. C., G. Schley, P. Liljeberg, M. Radetzki, J. Plosila, and H. Tenhunen (2013) "Optimal placement of vertical connections in 3D network-on-chip," *Journal of Systems Architecture*, **59**(7), pp. 441–454.

[127] Hsieh, A.-C. and T. Hwang (2012) "TSV redundancy: architecture and design issues in 3-D IC," *IEEE Transactions on Very Large Scale Integration Systems*, **20**(4), pp. 711–722.

[128] Kang, U., H.-J. Chung, S. Heo, S.-H. Ahn, H. Lee, S.-H. Cha, J. Ahn, D. Kwon, J.-H. Kim, J.-W. Lee, H.-S. Joo, W.-S. Kim, H.-K. Kim, E.-M. Lee, S.-R. Kim, K.-H. Ma, D.-H. Jang, N.-S. Kim, M.-S. Choi, S.-J. Oh, J.-B. Lee, T.-K. Jung, J.-H. Yoo, and C. Kim (2009) "8Gb 3D DDR3 DRAM using through-silicon-via technology," in *Proceedings of International Solid-State Circuits Conference.*

[129] Loi, I., F. Angiolini, S. Fujita, S. Mitra, and L. Benini (2011) "Characterization and implementation of fault-tolerant vertical links for 3-D networks-on-chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **30**(1), pp. 124–134.

[130] Hernandez, C., A. Roca, J. Flich, F. Silla, and J. Duato (2011) "Fault-tolerant vertical link design for effective 3D stacking," *Computer Architecture Letters*, **10**(2), pp. 41–44.

[131] Pasca, V., L. Anghel, C. Rusu, and M. Benabdenbi (2010) "Configurable serial fault-tolerant link for communication in 3D integrated systems," in *Proceedings of International On-Line Testing Symposium.*

[132] Cho, M. H., M. Lis, K. S. Shim, M. Kinsy, T. Wen, and S. Devadas (2009) "Oblivious Routing in On-Chip Bandwidth-Adaptive Networks," in *Proceedings of International Conference on Parallel Architectures and Compilation Techniques.*

[133] Dally, W. and B. Towles (2003) *Principles and Practices of Interconnect Networks*, Morgan Kaufmann Publishers Inc.

[134] Chao, C.-H., K.-Y. Jheng, H.-Y. Wang, J.-C. Wu, and A.-Y. Wu (2010) "Traffic- and thermal-aware run-time thermal management scheme for 3D NoC systems," in *Proceedings of International Symposium on Networks-on-Chip.*

[135] Dubois, F., A. Sheibanyrad, F. Ptrot, and M. Bahmani (2013) "Elevator-first: a deadlock-free distributed routing algorithm for vertically partially connected 3D-NoCs," *IEEE Transactions on Computers*, **62**(3), pp. 609–615.

[136] Jiang, N., D. Becker, G. Michelogiannakis, J. Balfour, B. Towles, D. Shaw, J. Kim, and W. Dally (2013) "A detailed and flexible cycle-accurate network-on-chip simulator," in *Proceedings of International Symposium on Performance Analysis of Systems and Software.*

[137] "Synopsys Design Compiler," `http://www.synopsys.com/Tools/Implementation/RTLSynthesis/DCGraphical/Pages/default.aspx`.

[138] Blech, I. A. and E. S. Meieran (1967) "Direct transmission electron microscope observation of electrotransport in Aluminum thin films," in *Proceedings of Reliability Physics Symposium.*

[139] Hau-Riege, C. S. (2004) "An introduction to Cu electromigraiton," *Microelectronics Reliability*, **44**(2), pp. 195 – 205.

[140] Black, J. (1969) "Electromigration–A brief survey and some recent results," *IEEE Transactions on Electron Devices*, **16**(4), pp. 338–347.

[141] Jing, J., L. Liang, and G. Meng (2010) "Electromigration simulation for metal lines," *Journal of Electronic Packaging*, **132**(1), p. 011002.

[142] Tu, K. (2011) "Reliability challenges in 3D IC packaging technology," *Microelectronics Reliability*, **51**(3), pp. 517–523.

[143] Zhang, T., K. Wang, Y. Feng, X. Song, L. Duan, Y. Xie, X. Cheng, and Y.-L. Lin (2010) "A customized design of DRAM controller for on-chip 3D DRAM stacking," in *Proceedings of Custom Integrated Circuits Conference.*

[144] Zhao, X., M. Scheuermann, and S. K. Lim (2012) "Analysis of DC current crowding in through-silicon-vias and its impact on power integrity in 3D ICs," in *Proceedings of Design Automation Conference.*

[145] Xu, Z., J.-Q. Lu, B. C. Webb, and J. U. Knickerbocker (2011) "Electromagnetic-SPICE modeling and analysis of 3D power network," in *Proceedings of Electronic Components and Technology Conference.*

[146] Li, D.-A., Z. Guang, M. Marek-Sadowska, and S. R. Nassif (2012) "Multi-via electromigration lifetime model," in *Proceedings of International Conference on Simulation of Semiconductor Processes and Devices.*

[147] Frank, T., S. Moreau, C. Chappaz, L. Arnaud, P. Leduc, A. Thuaire, and L. Anghel (2012) "Electromigration behavior of 3D-IC TSV interconnects," in *Proceedings of Electronic Components and Technology Conference.*

[148] Choi, H.-J., S.-M. Choi, M.-S. Yeo, S.-D. Cho, D.-C. Baek, and J. Park (2012) "An experimental study on the TSV reliability: electromigration and time dependant dielectric breakdown (TDDB)," in *Proceedings of International Interconnect Technology Conference.*

[149] Frank, T., C. Chappaz, P. Leduc, L. Arnaud, F. Lorut, S. Moreau, A. Thuaire, R. El Farhane, and L. Anghel (2011) "Resistance increase due to electromigration induced depletion under TSV," in *Proceedings of Reliability Physics Symposium.*

[150] Lin, M., N. Jou, J. Liang, and K. Su (2009) “Effect of multiple via layout on electromigration performance and current density distribution in copper interconnect,” in *Proceedings of Reliability Physics Symposium.*

[151] Zhong, Y. and M. Wong (2005) “Fast algorithms for IR drop analysis in large power grid,” in *Proceedings of International Conference on Computer-Aided Design.*

[152] Xie, J., V. Narayanan, and Y. Xie (2012) “Mitigating electromigration of power supply networks using bidirectional current stress,” in *Proceedings of Greak Lake Symposium on VLSI.*

[153] Todri, A., S.-C. Chang, and M. Marek-Sadowska (2007) “Electromigration and voltage drop aware power grid optimization for power gated ICs,” in *Proceedings of International Symposium on Low Power Electronics and Design.*

[154] Jain, P., T.-H. Kim, J. Keane, and C. H. Kim (2008) “A multi-story power delivery technique for 3D integrated circuits,” in *Proceedings of International Symposium on Low Power Electronics and Design.*

[155] Nassif, S. R. (2008) “Power grid analysis benchmarks,” in *Proceedings of Asia and South Pacific Design Automation Conference.*

[156] Healy, M. and S. K. Lim (2011) “A novel TSV topology for many-tier 3D power-delivery networks,” in *Proceedings of Design, Automation Test in Europe Conference Exhibition.*

[157] Zhang, C., V. Pavlidis, and G. De Micheli (2012) “Voltage propagation method for 3-D power grid analysis,” in *Proceedings of Design, Automation Test in Europe Conference Exhibition.*

[158] Ryu, S.-K., K.-H. Lu, X. Zhang, J.-H. Im, P. Ho, and R. Huang (2011) “Impact of near-surface thermal stresses on interfacial reliability of through-silicon vias for 3-D interconnects,” *IEEE Transactions on Device and Materials Reliability*, **11**(1), pp. 35–43.

[159] Pares, G., N. Bresson, S. Minoret, V. Lapras, P. Brianceau, J. F. Lugand, R. Anciant, and N. Sillon (2011) “Through silicon via technology using tungsten metallization,” in *Proceedings of International Conference on IC Design Technology.*

[160] Matsuoka, F., H. Iwai, K. Hama, H. Itoh, R. Nakata, T. Nakakubo, K. Maeguchi, and K. Kanzaki (1990) “Electromigration reliability for a tungsten-filled via hole structure,” *IEEE Transactions on Electron Devices*, **37**(3), pp. 562–568.

[161] "IBM 2011 Power Grid Simulation Contest," `http://dropzone.tamu.edu/~pli/PGBench`.

[162] Song, T., C. Liu, D. H. Kim, S.-K. Lim, J. Cho, J. Kim, J. S. Pak, S. Ahn, J. Kim, and K. Yoon (2011) "Analysis of TSV-to-TSV coupling with high-impedance termination in 3D ICs," in *Proceedings of International Symposium on Quality Electronic Design*.

[163] Kim, D. H., S. Mukhopadhyay, and S.-K. Lim (2011) "Fast and accurate analytical modeling of through-silicon-via capacitive coupling," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, **1**(2), pp. 168–180.

[164] Liu, C., T. Song, J. Cho, J. Kim, J. Kim, and S.-K. Lim (2011) "Full-chip TSV-to-TSV coupling analysis and optimization in 3D IC," in *Proceedings of Design Automation Conference*.

[165] Kaul, H., D. Sylvester, and D. Blaauw (2002) "Active shields: a new approach to shielding global wires," in *Proceedings of Great Lakes symposium on VLSI*.

[166] Subrahmanya, P., R. Manimegalai, V. Kamakoti, and M. Mutyam (2004) "A bus encoding technique for power and cross-talk minimization," in *Proceedings of International Conference on VLSI Design*.

[167] Victor, B. and K. Keutzer (2001) "Bus encoding to prevent crosstalk delay," in *Proceedings of International Conference on Computer Aided Design*.

[168] Duan, C., A. Tirumala, and S. Khatri (2001) "Analysis and avoidance of cross-talk in on-chip buses," in *Proceedings of Hot Interconnects*.

[169] Arunachalam, R., E. Acar, and S. Nassif (2003) "Optimal shielding/spacing metrics for low power design," in *Proceedings of Computer Society Annual Symposium on VLSI*.

[170] Li, L., N. Vijaykrishnan, M. Kandemir, and M. Irwin (2004) "A crosstalk aware interconnect with variable cycle transmission," in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition*.

[171] Chunjie Duan, S. P. K., Brock J. Lameres (2010) *On and off-chip crosstalk avoidance inVLSI design*, Springer.

[172] Chang, Y.-Y., Y.-C. Huang, V. Narayanan, and C.-T. King (2013) "ShieldUS: a novel design of dynamic shielding for eliminating 3D TSV crosstalk coupling noise," in *Proceedings of Asia and South Pacific Design Automation Conference*.

[173] Kumar, R. and S. P. Khatri (2013) "Crosstalk avoidance codes for 3D VLSI," in *Proceedings of Design, Automation Test in Europe Conference Exhibition*.

[174] Xu, Z., A. Beece, D. Zhang, Q. Chen, K.-N. Chen, K. Rose, and J.-Q. Lu (2010) "Crosstalk evaluation, suppression and modeling in 3D through-strata-via (TSV) network," in *Proceedings of International 3D Systems Integration Conference*.

[175] Stan, M. and W. Burleson (1995) "Bus-invert coding for low-power I/O," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **3**(1), pp. 49–58.

[176] Sotiriadis, P. and A. Chandrakasan (2000) "Low power bus coding techniques considering inter-wire capacitances," in *Proceedings of Custom Integrated Circuits Conference*.

[177] Leduca, P., F. de Crecy, M. Fayolle, B. Charlet, T. Enot, M. Zussy, B. Jones, J.-C. Barbe, N. Kernevez, N. Sillon, S. Maitrejean, and D. Louisa (2007) "Challenges for 3D IC integration: bonding quality and thermal management," in *Proceedings of International Interconnect Technology Conference*.

[178] Puttaswamy, K. and G. H. Loh (2006) "Thermal analysis of a 3D die-stacked high-performance microprocessor," in *Proceedings of Great Lakes Symposium on VLSI*.

[179] Das, S., A. Chandrakasan, and R. Reif (2004) "Timing, energy, and thermal performance of three-dimensional integrated circuits," in *Proceedings of Great Lakes Symposium on VLSI*.

[180] Jain, A., R. Jones, R. Chatterjee, and S. Pozder (2010) "Analytical and numerical modeling of the thermal performance of three-dimensional integrated circuits," *IEEE Transactions on Components and Packaging Technologies*, **33**(1), pp. 56–63.

[181] Wang, F., Z. Zhu, Y. Yang, and N. Wang (2011) "A thermal model for the top layer of 3D integrated circuits considering through silicon vias," in *Proceedings of International Conference on ASIC*.

[182] Chen, Y., E. Kursun, D. Motschman, C. Johnson, and Y. Xie (2011) "Analysis and mitigation of lateral thermal bloblock effect of through-silicon-via in 3D IC designs," in *Proceedings of International Symposium on Low-power Electronics and Design*.

[183] Athikulwongse, K., A. Chakraborty, J.-S. Yang, D. Pan, and S. K. Lim (2010) "Stress-driven 3D-IC placement with TSV keep-out zone and regularity study," in *Proceedings of International Conference on Autonomic Computing*.

[184] Jung, M., J. Mitra, D. Pan, and S. K. Lim (2011) "Tsv stress-aware full-chip mechanical reliability analysis and optimization for 3D IC," in *Proceedings of Design Automation Conference*.

[185] Lu, K.-H., S.-K. Ryu, J. Im, R. Huang, and P. S. Ho (2011) "Thermo-mechanical reliability of through-silicon vias in 3D interconnects," in *Proceedings of International Reliability Physics Symposium*.

[186] Selvanayagam, C. S., J. H. Lau, X. Zhang, S. Seah, K. Vaidyanathan, and T. C. Chai (2009) "Nonlinear thermal stress/strain analyses of copper filled TSV (through silicon via) and their flip-chip microbumps," *IEEE Transactions on Advanced Packaging*, **32**(4), pp. 720–728.

[187] Lu, K. H., X. Zhang, S.-K. Ryu, J. Im, R. Huang, and P. S. Ho (2009) "Thermo-mechanical reliability of 3-D ICs containing through silicon vias," in *Proceedings of Electronic Components and Technology Conference*.

[188] Noritake, C., P. Limaye, M. Gonzalez, and B. Vandevelde (2006) "Thermal cycle reliability of 3D chip stacked package using Pb-free solder bumps: parameter study by FEM analysis," in *Proceedings of International Conference on Thermal, Mechanical and Multiphysics Simulation and Experiments in Micro-Electronics and Micro-Systems*.

[189] Hung, W.-L., G. Link, Y. Xie, N. Vijaykrishnan, and M. Irwin (2006) "Interconnect and Thermal-Aware Floorplanning for 3D Microprocessors," in *Proceedings of International Symposium on Quality Electronic Design*.

[190] Huang, W., S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. Stan (2006) "HotSpot: A Compact Thermal Modeling Methodology for Early-Stage VLSI Design," *IEEE Transactions on Very Large Scale Integration Systems,*, **14**(5), pp. 501–513.

[191] Chen, Z., X. Luo, and S. Liu (2010) "Thermal analysis of 3D packaging with a simplified thermal resistance network model and finite element simulation," in *Proceedings of International Conference on Electronic Packaging Technology High Density Packaging*.

[192] David, H., C. Fallin, E. Gorbatov, U. R. Hanebutte, and O. Mutlu (2011) "Memory Power Management via Dynamic Voltage/Frequency Scaling," in *Proceedings of International Conference on Autonomic Computing*.

[193] Goplen, B. and S. Sapatnekar (2005) "Thermal via placement in 3D ICs," in *Proceedings of International Symposium on Physical Design.*

Vita

Qiaosha Zou

Qiaosha Zou received her B.S. degree from Department of Computer Science and Technology of Huazhong University of Science and Technology in China. She also received a Bachelor's degree of Economics from Wuhan University in China. She was admitted into Pennsylvania State University in 2009. Then She joined the Ph.D. program in Department of Computer Science as a member of Microsystems Design Laboratory (MDL) group working with Professor Yuan Xie in 2010. Her research interests includes the design automation in three-dimensional integrated circuits (3D ICs) and computer architecture designs with emerging technologies. She has published and co-authored 11 conference papers and one book chapter. She has also served as a peer reviewer for journals and conferences in the field of computer architecture and electronic design automation, including Design Automation Conference (DAC), International Conference on Computer Aided Design (ICCAD), ACM Journal on Emerging Technologies in Computing Systems (JETC), and Journal of Circuits, Systems, and Computers , etc.