RELIABLE CLOCK AND POWER DELIVERY NETWORK DESIGN FOR THREE-DIMENSIONAL INTEGRATED CIRCUITS

A Dissertation
Presented to
The Academic Faculty

by

Xin Zhao

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

Georgia Institute of Technology
December 2012

RELIABLE CLOCK AND POWER DELIVERY NETWORK DESIGN FOR THREE-DIMENSIONAL INTEGRATED CIRCUITS

Approved by:

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

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

Dr. Muhannad Bakir
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Madhavan Swaminathan
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Hyesoon Kim
College of Computing
Georgia Institute of Technology

Date Approved: October 12, 2012

Dedicated to my beloved family:

To my parents, Deshan Zhao and Chunying Liu

to my husband, Hongyi Qu

and to my son, Kevin Qu

for boundless love, support, and encouragement.

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my advisor, Professor Sung Kyu Lim for his guidance, professional advice, and support through my Ph.D. study at Georgia Tech. I would also like to express my sincere thanks to Dr. Michael R. Scheuermann from IBM T. J. Watson Research Center. I am grateful to have had the chance to learn and develop under his expert guidance. I would like to thank Professor Saibal Mukhopadhyay for numerous discussions that provide insights on my research and for serving as my proposal committee and dissertation reading committee. I thank Professor Hsien-Hsin S. Lee for providing in-depth discussions on my research and serving as my proposal committee. I am thankful to Professor Muhannad Bakir, Professor Madhavan Swaminathan, and Professor Hyesoon Kim for serving as my dissertation committee and providing useful feedbacks. I am also grateful to Dr. Gabriel H. Loh.

I would like to extend my thanks to the GTCAD members (past and present) for insightful discussions, valuable comments, and friendship: Dr. Michael B. Healy, Mohit Pathak, Dr. Dae Hyun Kim, Dr. Krit Athikulwongse, Young-Joon Lee, Moongon Jung, Taigon Song, Chang Liu, Shreepad Panth, Yang Wan, Steven Zhang, Yarui Peng, Woongrae Kim, and Sandeep Samal. I would also like to thank GREEN, MARS, STING, and Epsilon members for sharing knowledge and collaborations: Dr. Jeremy R. Tolbert and Dr. Dean L. Lewis for practical discussions and close collaboration, Subho Chatterjee, Minki Cho, Amit R. Trivedi, Kwanyeob Chae, Dr. Dong Hyuk Woo, Tzu-Wei Lin, Mohammad M. Hossain, Guanhao Shen, and Jianyong Xie.

I gratefully acknowledge the support of the VLSI design team in IBM. They provided me an internship, which allowed me to broaden my knowledge in reliability issues.

I express my deepest gratitude to my beloved parents Deshan Zhao and Chunying Liu for everything they have provided me. A special word of thanks must go to my husband, Hongyi Qu, for his endless devotion and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The main objective of this thesis is to design reliable clock-distribution networks and power-delivery networks for three-dimensional integrated circuits (3D ICs) using through-silicon vias (TSVs). This dissertation supports this goal by addressing six research topics. The first four works focus on 3D clock tree synthesis for low power, pre-bond testability, TSV-induced obstacle avoidance, and TSV utilization. The last two works develop modeling approaches for reliability analysis on 3D power-delivery networks.

In the first work, a clock synthesis algorithm is developed for low-power and low-slew 3D clock network design. The impact of various design parameters on clock performance, including the wirelength, clock power, clock slew, and skew, is investigated. These parameters cover the TSV count, TSV parasitics, the maximum loading capacitance of the clock buffers, and the supply voltage.

In the second work, a clock synthesis algorithm is developed to construct 3D clock networks for both pre-bond testability and post-bond operability. Pre-bond testing of 3D stacked ICs involves testing each individual die before bonding. The overall yield of 3D ICs improves with pre-bond testability because manufacturers can avoid stacking defective dies with good ones. Two key techniques including TSV-buffer insertion and redundant tree generation are implemented to minimize clock skew and ensure pre-bond testing. The impact of TSV utilization and TSV parasitics on clock power is also investigated.

In the third work, an obstacle-aware clock tree synthesis method is presented for through-silicon-via (TSV)-based 3D ICs. A unique aspect of this problem lies in the fact that various types of TSVs become obstacles during 3D clock routing including signal, power/ground, and clock TSVs. These TSVs may occupy silicon area or routing layers. The generated clock tree does not sacrifice wirelength or clock power too much and avoids TSV-induced obstacles.

In the fourth work, a decision-tree-based clock synthesis (DTCS) method is developed for low-power 3D clock network design, where TSVs form a regular 2D array. This TSV array style is shown to be more manufacturable and practical than layouts with TSVs located at irregular spots. The DTCS method explores the entire solution space for the best TSV array utilization in terms of low power. This method is applied for both gate-level chip-scale 3D clock designs and block-level global clock designs. Close-to-optimal solutions can be found for power efficiency with skew minimization in short runtime.

In the fifth work, current crowding and its impact on 3D power grid integrity is investigated. Due to the geometry of TSVs and connections to the global power grid, significant current crowding can occur. The current density distribution within a TSV and its connections to the global power grid is explored. A simple TSV model is implemented to obtain current density distributions within a TSV and its local environment. These models are checked for accuracy by comparing with identical models simulated using finite element modeling methods. The simple TSV models are integrated with the global power wires for detailed chip-scale power analysis.

In the sixth work, a comprehensive multi-physics modeling approach is developed to analyze electromigration (EM) in TSV-based 3D connections. Since a TSV has regions of high current density, grain boundaries play a significant role in EM dominating atomic transport. The transient analysis is performed on atomic transport including grain and grain boundary structures. The evolution of atomic depletion and accumulation is simulated due to current crowding. And the TSV resistance change is modeled.

CHAPTER I

INTRODUCTION AND BACKGROUND

1.1 Introduction

Three-dimensional integrated circuits (3D ICs) have gradually shown promising potentials of low cost, further miniaturization, small area, low power, high bandwidth, and heterogeneous stacking enabled [2–5]. In 3D ICs, the clock distribution network spreads over the entire stack to distribute the clock signal to all the sequential elements. Clock skew, defined as the maximum difference in the clock signal arrival times from the clock source to all sinks, is required to be less than 3% or 4% of the clock period in an aggressive clock network design according to the International Technology Roadmap for Semiconductors (ITRS) projection [6]. Thus, clock skew control, which was well studied in 2D ICs [7], is still a primary objective in the 3D clock network design.

The clock signal in 3D ICs is distributed not only along the X and Y directions, but also along the Z direction using through-silicon vias (TSVs). The clock distribution network drives large capacitive loads and switches at a high frequency, which leads to an increasingly large proportion of the total power dissipated in the clock distribution network. In some applications, the clock network itself is responsible for 25% [8] and even up to 50% [9] of the total chip power consumption. Moreover, because a large clock slew may cause a setup or hold time violation, the clock slew must also be taken into consideration when designing a 3D clock network. Thus, low power, skew, and slew remain important design goals in 3D clock networks.

For a reliable 3D clock network design, several challenging issues should be taken care. First, in-depth investigations on the impact of TSV utilization on clock power and performance is important. This study can help designers understand the policies of robust 3D clock designs and apply efficient techniques accordingly. Second, the pre-bond testing [10], which tests each individual die before bonding, is able to improve the overall yield of 3D

ICs by avoiding stacking defective dies with good ones. 3D clock designs should ensure both pre-bond testability and post-bond operability with minimum skew and low power consumption. Third, TSVs may occupy silicon area or routing layers, which are the obstacles during 3D clock routing. 3D clock trees should avoid overlapping with TSV-induced obstacles and should not sacrifice wirelength or clock power too much. Fourth, the TSV array style, where TSVs form a regular 2D array, is shown to be more manufacturable and practical than layouts with TSVs located at irregular spots. The utilization of TSV arrays for 3D clock synthesis significantly affects the clock power. An automatic and efficient approach, which can find an optimum solution of the low-power clock design, is required for TSV array design style.

In addition, electromigration (EM) has been studied for many decades and is still an open issue as an unavoidable source of degradation [11–13]. Voids in the conductive material can grow over time and may result in an open-circuit failure. A few studies have been presented on TSV EM modeling and analysis [14,15]. However, none of them investigated the detailed current distribution inside TSVs and the resulting thermal and stress migrations, where some of the corners may have the large current gradient and suffer EM reliability issues. Furthermore, a comprehensive multi-physics modeling approach is essential for designers to better understand the EM phenomenon and improve the EM lifetime in TSV-based 3D connections.

The reliable power network design is also a critical factor for robust circuit performance. The supply voltage scales slower than the scaling trend of transistors and interconnects. The increased current density and temperature accelerate the transistor and wire degradation, and shorten the lifetime of electronic devices [12]. Therefore, the 3D power integrity analysis for EM reliability is important to reliable 3D integration.

1.1.1 Contributions

The contributions of this thesis are summarized as follows:

- **A comprehensive clock synthesis algorithm for 3D ICs**: A two-step approach is developed, which includes (1) three-dimensional (3D) abstract tree generation based on the three-dimensional method of means and medians (3D-MMM) algorithm and (2) buffering and embedding based on the slew-aware deferred-merge buffering and embedding (sDMBE) algorithm. In addition, an extension of the 3D-MMM method (3D-MMM-ext) is implemented to determine the optimal number of TSVs in the 3D clock tree and to minimize the overall power consumption. This 3D-MMM-ext method can find a close-to-optimal design point in the "TSV count vs. power consumption" tradeoff curve very efficiently.

- **An in-depth investigation on the impact of TSV utilization on 3D clock performance**: For the first time, an extensive investigation on the impact of the TSV count and the TSV parasitics on clock power consumption and performance is presented. Several techniques are introduced to reduce the clock power consumption and clock slew of the 3D clock-distribution network. We analyze how these design factors affect the overall wirelength, clock power, slew, and skew in the clock network designs. Two important observations are made: (1) A 3D clock network that uses multiple TSVs significantly reduces the clock power compared with the single-TSV case; and (2) as the TSV capacitance increases, the power savings of a multiple-TSV clock network decreases.

- **The first clock design methodology for pre-bond testing in 3D ICs**: For the first time, a 3D clock synthesis methodology and algorithm for pre-bond testing is developed and implemented. Two key techniques including TSV-buffer insertion and redundant tree generation are implemented to minimize clock skew and ensure pre-bond testing. The impact of TSV utilization and TSV parasitics on wirelength and clock power is also investigated. Compared with the single-TSV solution, the proposed method minimizes the overall wirelength, reduces clock power consumption, and provides both pre-bond testability and post-bond operability with minimum skew and constrained slew.

- **The first clock synthesis algorithm for TSV-induced obstacle avoidance**: For the first time, a comprehensive analysis on TSV-induced obstacles is performed and a clock routing algorithm for TSV-induced obstacle avoidance is developed and implemented. The traditional concept of merging segment is extended to represent the clock TSV and clock buffer insertion. Two key techniques are developed to determine overlap-free merging segments including Expanded-Obstacle Cutting and Nine-Region-Based Cutting techniques; two detour policies are presented to handle clock routing in heavily crowded regions. This proposed method can generate 3D clock trees that do not sacrifice wirelength or clock power too much and avoid overlapping with TSV-induced obstacles.

- **The first clock synthesis algorithm of TSV array utilization for low-power 3D clock design**: For the first time, an efficient clock synthesis methodology for TSV array design style is presented. A decision-tree-based clock synthesis (DTCS) method is developed for low-power 3D clock network design, where TSVs from a regular 2D array. The DTCS method explores the entire solution space for the best TSV array utilization in terms of low power. This method is applied for both gate-level chip-scale 3D clock designs and block-level global clock designs. Close-to-optimal solutions can be found for power efficiency with skew minimization in short runtime.

- **A detail investigation on current density distribution in the TSV-to-wire interface and a TSV model for chip-scale 3D power integrity analysis**: The current density distribution within a TSV and its connections to the global chip power grid is explored, where a significant amount of current crowding is observed. Simple TSV models are implemented to obtain current density distributions within a TSV and its local environment. These models are checked for accuracy by comparing with identical models simulated using finite element modeling methods. This simple TSV models are integrated with the 3D global power-delivery networks for detailed chip-scale power analysis.

- **The first multi-physics modeling approach for transient analysis on Electromigration in TSV-based 3D connections**: For the first time, a multi-physics modeling approach for electromigration analysis is presented for TSV-based three-dimensional connections, where transient analysis is performed on atomic transport and TSV effective resistance including grain and grain boundary structure. The evolution of atomic depletion and accumulation is simulated due to current crowding. The model is validated by exploring the impact of current, temperature, and various grain sizes on the EM reliability. In addition, the TSV effective resistance evolution is modeled. These results and discussions provide guidance for designers to better understand and avoid EM reliability failures in 3D ICs.

1.1.2 Thesis Organization

This dissertation is organized as follows:

- Chapter 1 introduces the thesis of this dissertation, summarizes the contributions, and explains the organization of this dissertation.
- Chapter 2 presents a clock synthesis algorithm for low-power 3D clock network designs and investigates the TSV utilization on clock power reduction.
- Chapter 3 presents a clock synthesis algorithm of 3D clock network design for pre-bond testability.
- Chapter 4 describes an obstacle-aware clock tree synthesis method for TSV-based 3D clock networks.
- Chapter 5 provides a clock synthesis method of TSV-array utilization for low-power 3D clock network design.
- Chapter 6 investigates current crowding in TSV-to-wire interfaces and develops a simple TSV model for chip-scale 3D power-integrity analysis.
- Chapter 7 presents a comprehensive multi-physics modeling approach to analyze electromigration in TSV-based 3D connections.
- Chapter 8 summarizes the research presented in this dissertation and provides concluding remarks.

1.2 Background

1.2.1 Traditional Clock Network Design

The clock distribution network plays an important role in a synchronous digital system. Clock skew, defined as the maximum difference in the clock signal arrival times, can severely limit the maximum performance of the entire system. The clock skew is required to be less than 3% or 4% of the clock period in an aggressive clock network design according to ITRS projection [6]. The clock distribution network travels the entire chip, drives large capacitive loads, and operates at a high frequency [7, 8]. In some applications, the clock network can consume 25% [8] and even up to 50% [9] of the total chip power. In addition, the clock network is sensitive to the thermal gradients, process variations, and systematic variations.

The clock tree construction was first implemented as a H-tree [16, 17]. This symmetric structure significantly reduces clock skew but generates large size of the clock network. Jackson et al. [18] presented robust clock routing techniques for high-performance VLSI circuits. Their algorithm, called the method of means and medians (MMM), generates a clock topology by recursively partitioning the sink set into two subsets and then connecting the centers of these sets. Cong et al. [19] proposed a bottom-up matching approach to construct clock trees and addressed clock skew minimization in the linear delay model. Tsay [20] constructed clock trees with exact zero skew under the Elmore delay model [21]. Chao et al. [22] presented the deferred-merge embedding (DME) algorithm, which achieved shorter wirelength than both the MMM algorithm [18] and the bottom-up matching algorithm [19].

Process variation is a critical aspect of semiconductor fabrication [23]. Several work focused on analyzing the variation impact on clock networks [24–26]. Sauter et al. [24] compared four clock topologies in the presence of die-to-die (D2D) and within-die (WID) process variations, including a H-tree, a clock network with interleaved rings, a trunk tree, and a clock grid. Narasimhan et al. [25] analyzed the process variation impact on a five-stage 2D H-tree in various technology nodes.

Many clock synthesis algorithms have been proposed to improve the variation robustness in a clock network. Early in 1996, Neves and Friedman [27] proposed a clock design methodology to tolerate process parameter variations. Padmanabhan et al. [28] developed a statistical centering-based clock routing technique for the DME algorithm. Lam and Koh [29] integrated the clock scheduling and clock routing to tolerate process variations. Venkataraman et al. [30] addressed the same issue in various stages of the clock synthesis, including skew scheduling, abstract tree generation, and embedding. Rajaram et al. [31] developed a non-tree clock network. They inserted crosslinks in a given clock tree and analyzed the clock skew variation caused by crosslink insertion.

Several studies have been proposed to address low-power clock network design for high-performance and reliable VLSI systems. Dynamic-programming-based buffer insertion mainly focused on wirelength-driven, timing-driven, and maximum-slew-driven designs with power or area minimization [32–37]. Wang et al. [38] formulated the wire and buffer sizing problem as a sequential linear programming problem to minimize clock power under the skew constraint. Guthaus et al. [39] solved the clock tree sizing problem with the consideration of process variations. They minimized the process-variation-aware skew under the given power budget. Cho et al. [40] constructed a clock tree to balance the clock skew under two given static thermal profiles. Chakraborty et al. [41] extended the study [40] by considering the bounded clock skew. Moreover, Yu et al. [42] constructed thermal-aware clock trees by computing many bottom-up merging points based on the thermal sensitivity.

1.2.2 Clock Network Design in Three-Dimensional ICs

The history of clock network design for 3D stacked ICs is short. Pavlidis et al. [43] presented measurement data from a fabricated 3D clock distribution network. Arunachalam and Burleson [44] used a separate layer for the clock distribution network to reduce power. Minz et al. [45] proposed the first work on the 3D clock synthesis and studied the clock skew minimization with the impact of the thermal gradient. The clock topology consists of a complete clock tree in one die and many subtrees in other dies. Their results showed a significant wirelength reduction using many TSVs. Xu et al. [46] proposed a statistical clock skew model for the 3D H-tree design.

Though 3D stacked ICs offer potential attractions, the success of 3D ICs is predicated on the final post-bond yield. Lee and Chakrabarty [47] presented a comprehensive study on the challenges of testing 3D ICs. Marinissen and Zorian [48] provided an overview of manufacturing processes in TSV-based 3D stacked ICs and discussed the test challenges. To reduce the chance of bonding good dies to the defective ones, each die should be tested prior to the bonding process. Lewis and Lee presented an architectural solution [10] to the pre-bond testability problem for 3D die-stacked microprocessors. They discussed how to perform testing for functional modules that are partitioned across multiple dies. They also investigated new design and test methods [49] to address similar issues for 3D circuits. Jiang et al. [50] presented a heuristic method to optimize the test time and routing cost for both post-bond test and pre-bond wafer-level test. In addition, Jiang et al. [51] developed a technique test-architecture design technique under a constrained pre-bond test pin count.

In TSV-based 3D ICs, TSVs create serious blockages for 3D clock routing. Before clock tree synthesis, P/G TSVs and signal TSVs are inserted and occupy both silicon and metal space. TSVs are significant layout obstacles due to their large size compared with logic gates and local wires. Clock routing in 3D IC becomes challenging because these various types of TSVs all become obstacles. In existing works on obstacle-aware clock routing, Kahng and Tsao [52] proposed deferred merging and embedding (DME)-based obstacle expansion rules to determine feasible embedding locations for the internal nodes. In [53], Kim and Zhou presented a planar obstacle-aware routing scheme to clean up overlaps between clock nets and obstacles. Huang *et al.* [54] proposed another DME-based clock routing method to avoid obstacles with the help of a track graph. These works mainly focus on routing-obstacle avoidance, *i.e,* to prevent clock nets from crossing over the given obstacles. In addition, there are several works on avoiding insertion of clock buffers on the given blockages based on either maze routing [55] [56] or breadth-first-search [57]. However, none of these work can directly solve the TSV-obstacles in 3D clock tree construction problem.

1.2.3 Reliability Issues in TSVs

Through-silicon vias (TSVs) may cause reliability and cost issues that delay mainstream acceptance [4,58]. TSVs can squeeze or stretch adjacent transistors and interconnects. This material deformation may lead to mobility change and thus performance variation [4,59].

It also causes mechanical reliability issues, causing open hole, short, or even crack. TSV-to-TSV and TSV-to-device coupling affects timing and signal integrity [60–62]. All these TSV-related issues require extra design efforts.

The TSV array, defined as a group of TSVs placed in regular positions either in one-dimensional or two-dimensional grid fashion, is shown to be more manufacturable and practical to address the TSV-related reliability issues. Recent studies show that placing TSVs at any desired locations during placement [63] or routing [64] leads to shorter wirelength and better timing results compared with the regular locations (TSV arrays). However, this irregular placement may result in TSVs crowded in a certain region and cause problems in coupling [61, 65], timing variations [62, 66], and mechanical reliability [59, 67].

Electromigration (EM) decreases the reliability of integrated circuits (ICs). It may eventually cause shorts or opens in circuits and interconnects which can reduce IC lifetimes, or worst, cause field fails. EM is driven by multiple physical mechanisms, including electric current, temperature gradient, stress gradient, and atomic concentration gradient. The evolution of atomic concentration or the mean time to failure (MTTF) are two important parameters to investigate the EM reliability. This analysis requires a transient analysis of the atomic concentration. Atomic diffusion is significantly different within a metal grain and along grain boundaries, each having different activation energies. Atomic transport is dominated by grain boundary diffusion and must be included in any realistic EM simulation.

EM modeling and analysis for interconnects have been extensively studied for many decades [68, 69]. However, the history of modeling on TSV reliability is very short. A recent paper analyzed and modeled the DC current crowding inside the TSVs and at the connections between TSVs and power wires [70]. This crowding increases the TSV effective resistance and voltage drop in the power-delivery network of 3D ICs. Some papers modeled the thermal-mechanical stress at the interface between TSVs and the substrate [67, 71] and its impact on device performance [62]. The impact of TSV stress on back end of line (BEOL) interconnects and its EM lifetime was modeled and discussed [14, 72]. A modeling approach on TSV EM reliability was also proposed in [15]. However, none of these works present transient analysis on atomic concentration for EM lifetime.

In addition, few papers [73, 74] include grain and grain boundary simulation. However, all of these works discussed one-dimensional wires where current density is fairly uniform, not three-dimensional connections. TSVs, which typically have a high average current density, can have much higher local current densities due to current crowding. These regions of high local current density are much more susceptible to EM degradation. Moreover, the large power density with high temperature or large thermal gradient inside 3D ICs due to multi-tier stacking or joule heating can accelerate atomic migration. Therefore, analyzing the evolution of atomic concentration and the EM lifetime for the 3D connection is important.

1.2.4 3D Power Integrity Analysis for EM Reliability

Power-delivery network (PDN) design has become a challenging task in ICs as technology scales. Since the supply voltage scales slower than transistors and interconnects, the current density has been rapidly increasing. The increased current density, along with the high temperature, accelerates transistor and wire degradation and shortens the lifetime of both devices and wires. Today, the current density can reach to several hundred thousands of amperes per square centimeter. At this current density magnitude, electromigration (EM) becomes significant. PDN design needs to be accurately checked for excessive current density to insure EM limits are not exceeded and voltage drops (IR) are within specifications before releasing to manufacturing.

EM and IR drop problems are compounded for 3D ICs. Specifically, a 3D PDN provides power supply to all devices in the entire 3D stack. The inter-die power-delivery interconnects, formed by power/ground (P/G) through-silicon-vias (TSVs) or micro-bumps, are unique components in 3D power grids. These vertical connections carry large amounts of current and may suffer from EM degradation due to an excessive current density as well as have large IR drops. Therefore, detailed and accurate analysis on the 3D PDN is important to predict the performance and improve the power integrity as necessary.

Some recent papers discussed TSV EM modeling and analysis [14, 15] and TSV-based 3D PDN analysis [75]. However, none of these works investigates detailed current density distribution or current crowding inside P/G TSVs, where some of the edges may suffer from a large current gradient and are subject to a potential EM reliability issue. Moreover, prior works model TSVs and powers wire segments as single resistors, which are insufficient to accurately analyze the detailed current density distribution inside P/G TSVs and 3D PDNs.

CHAPTER II

LOW-POWER CLOCK NETWORK DESIGN FOR 3D ICS

In three-dimensional integrated circuits (3D ICs), TSVs provide the vertical interconnections to deliver the clock signal to all dies in the 3D stack. The low-power 3D clock network design requires a thorough investigation on how the TSV count and TSV parasitics affect the clock performance. Existing work has demonstrated that the total wirelength of a 3D clock network decreases significantly if more TSVs are used [45, 76–78]. According to the observations made in [45], the die that contains the clock source includes a complete tree, while other dies can have subtrees, as illustrated in Figure 1.

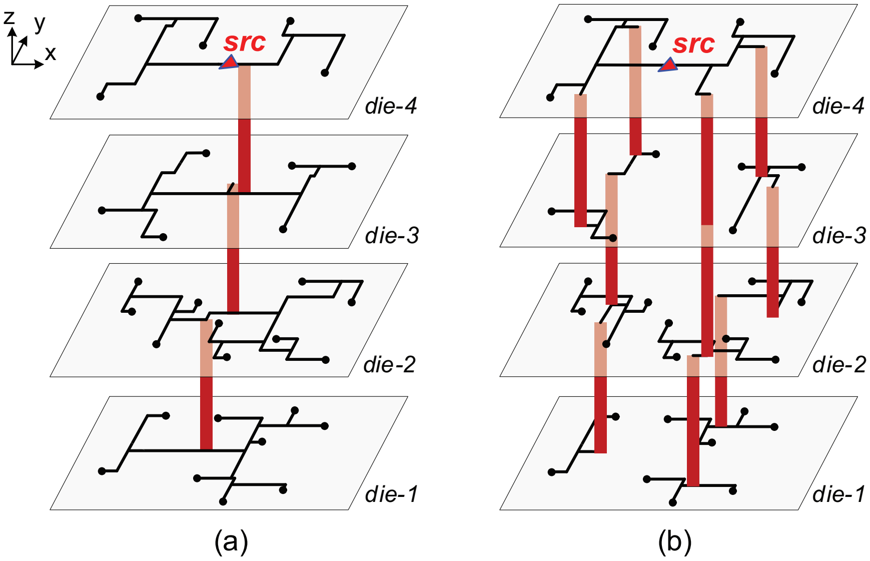


Figure 1: Four-die stack 3D clock networks with two different TSV counts. (a) uses single TSV between adjacent dies; (b) uses ten TSVs. The overall wirelength is shorter in (b).

A 3D clock tree that utilizes multiple TSVs tends to reduce the overall wirelength as more and more TSVs are used. However, the analysis of TSV RC parasitics on the clock network has not been addressed in the literature. If a 3D clock tree utilizes many TSVs that have large TSV RC parasitics, the clock delay and power consumption contributed by the TSVs may increase significantly. Using more TSVs helps to reduce the wirelength and thus power consumption, but the TSV capacitance increases the clock power consumed at the same time.

In this chapter, an in-depth investigation is performed on the impact of various design parameters on the wirelength, clock power, slew, and skew of the 3D clock network. These parameters include the total clock TSV count, the TSV parasitics, the maximum loading capacitance of the clock buffers, and the supply voltage. The "TSV count vs. clock power" tradeoff curves are generated for various TSV parasitic values. The TSV count and the TSV capacitance effect on clock power is discussed. Using multiple TSVs helps to reduce the maximum and average slew compared with the single-TSV case. An effective approach to determine the optimal number of TSVs is presented for the 3D clock tree so that the overall power consumption is minimized. This method predicts the impact of adding a new TSV into the current clock topology on the overall power consumption during the top-down abstract tree generation. This prediction helps to decide whether pairing of two clock nodes in different dies and using a TSV for this pair is useful for power reduction or not. A close-to-optimal design point can be determined in the TSV count vs. power consumption tradeoff efficiently compared with a straightforward exhaustive search method.

2.1 Preliminaries

2.1.1 Electrical and Physical Model of 3D Clock Network

A 3D clock network is modeled as a distributed resistance and capacitance (RC) network. The sink nodes that represent flip-flops and clock input pins of memory blocks are modeled as capacitive loads. The wire segments and TSVs are represented as π models[1], which is a classical way to represent the parasitics of a clock network. Each buffer or driver is constructed with two inverters. Note that prior works have focused on the electrical modeling of TSVs [65, 79–81]. Our 3D clock routing algorithm is flexible to handle more complicated TSV parasitic models than the lumped model.

The TSV bound is a constraint on the maximum TSV number for each die. The TSV

[1] In this work, wire segments denote the edges of the abstract tree and are not uniformly distributed. Depending on the TSV insertion and buffer insertion on the abstract tree, a src-to-sink path usually contains tens of wire segments, where each segment length varies from tens of micrometers to a few hundreds of micrometers.

bound is usually decided before clock synthesis. Different from the TSV bound, the TSV count (#TSVs) is the total number of TSVs utilized in the 3D clock tree. For an n-die 3D stack, #TSVs is usually less than or equal to (n-1) times the TSV bound.

A three-die clock interconnect using four TSVs is shown in Figure 2: The clock source is located in die-3; sink a in die-1 connects to the source using two vertically aligned TSVs; sink c in die-1 connects to the source by two TSVs; and sink b in die-2 uses one TSV.

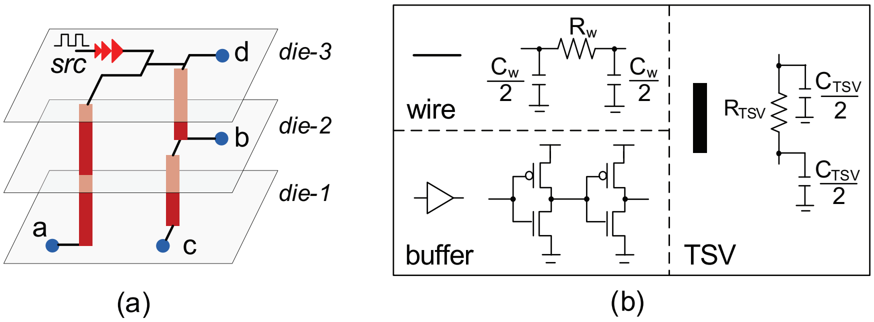


Figure 2: A sample clock tree and its electrical model. (a) A sample three-die clock network using four TSVs. The clock source is in die-3. Sink a in die-1 uses two vertically aligned TSVs. And Sink b in die-2 uses one TSV to connect to the clock source. (b) Electrical models of the clock wire segments, TSVs, and buffers/drivers.

2.1.2 Problem Formulation

Given a set of sinks in all dies, a TSV bound, a pre-determined clock source location, and the parasitics of wires, buffers, and TSVs, the 3D clock synthesis constructs a fully-connected 3D clock network satisfying the following conditions: (1) Clock sinks in all dies are connected by a single tree; (2) the TSV count in each die is under the TSV bound; (3) the clock skew is minimized; (4) the clock slew is below the constraint; and (5) the wirelength and clock power are minimized.

Clock skew is the maximum difference among the arrival times at the clock sinks. In the existing clock synthesis tools, the Elmore delay model is widely used for RC delay and skew calculation. The primary goal of our 3D clock synthesis is to construct a zero-Elmore-skew clock network. The SPICE simulation is performed to achieve the accurate timing information and to evaluate the clock synthesis performance. The simulated clock skew is constrained to less than three percent of the clock period. The clock slew is defined as the transition time from 10% to 90% of the clock signal at each sink.

The TSV bound constraint plays an important role in achieving low-power 3D clock networks. This constraint reflects the impact of the TSV usage on routing congestion, capacitive coupling, and stress-induced manufacturing issues. By varying the TSV bounds, we obtain different 3D clock networks. Note that the TSV bound is different to the actual TSV usage in each die, because this bound only limits the maximum TSV usage for each die.

2.2 3D Clock Tree Synthesis

2.2.1 Overview

The 3D clock synthesis algorithm consists of two major steps: (1) 3D abstract tree generation and (2) slew-aware buffering and embedding. A 3D abstract tree is generated based on the 3D method of means and medians (3D-MMM) algorithm. The 3D-MMM algorithm determines the connections of nodes (sink nodes or merging points) and uses TSVs if necessary. Note that the 3D-MMM algorithm works in such a way that the sinks in one die are connected by a *single tree*, whereas the sinks in other dies are connected by *multiple trees*. The clock source is located in the die that contains the single tree.

Once a 3D abstract tree is obtained, we determine the routing topology and exact geometric locations for all the nodes, TSVs, and buffers. Our slew-aware deferred-merge buffering and embedding (sDMBE) method is a two-phase approach, which is based on the classic deferred-merge and embedding (DME) algorithm [22] for clock routing. The sDMBE method first visits each node in a bottom-up fashion, determines the merging type for a pair of subtrees, inserts buffers if necessary, and calculates the merging distances based on the zero-Elmore-skew equations. The outcomes after the first phase are the merging segments, which store the feasible locations of the internal nodes in the 3D abstract tree. In the second phase, the sDMBE method visits the whole abstract tree in a top-down manner while deciding the exact merging locations for the internal nodes, buffers, TSVs, and exact routing topology. All the sinks are connected in a single tree.

2.2.2 3D Abstract Tree Generation

The first step of our 3D clock synthesis is the 3D abstract tree generation using the 3D-MMM algorithm. A 3D abstract tree indicates the hierarchical connection information among the sink nodes, internal nodes, TSVs, and the root node. The 3D abstract tree of an n-die stack clock network is an n-colored binary tree that identifies the die indices for all the nodes.

We develop the 3D-MMM algorithm to generate a 3D abstract tree for the given clock sinks in a top-down manner, which is an extension of the method of means and medians (MMM) algorithm [18]. The 3D abstract trees generated by the 3D-MMM algorithm with various TSV bounds are shown in Figure 3. Note that a larger TSV bound moves TSVs closer to the sink nodes and causes more vertical clock connections than horizontal connections. However, the overall wirelength is reduced because of the short horizontal connection length.

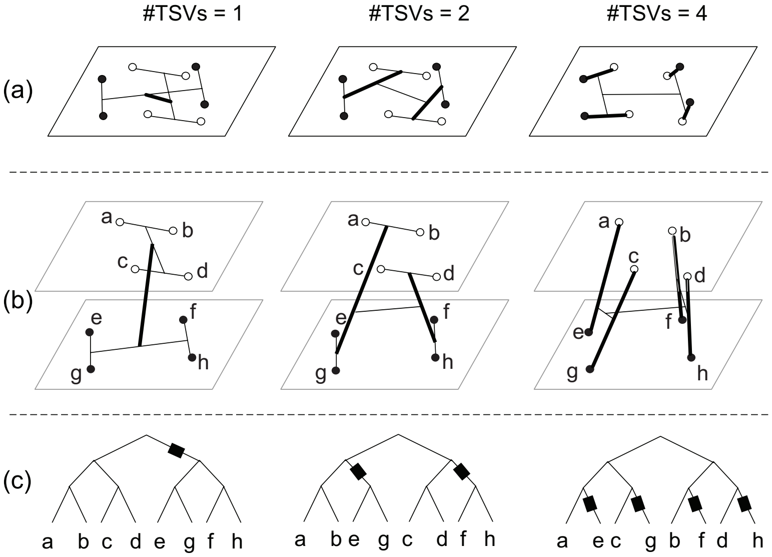


Figure 3: The 3D abstract trees generated by the 3D-MMM algorithm under various TSV bounds. (a) 2D view, where thick lines denote TSV connection. (b) 3D view. (c) Binary abstract trees, where the squares denote TSVs.

The basic idea of our 3D-MMM algorithm is to recursively divide the given sink set into two subsets until each sink belongs to its own set. A TSV is used if we decide to merge a pair of nodes in different dies. Our goal is to evenly distribute the TSVs across the die area under the given TSV bound. This even TSV distribution is shown to improve manufacturability [62].

Let $S = \{s_1, s_2, .., s_k\}$ denote a set of sinks, where the locations of the sinks have been decided before the 3D clock tree synthesis. We assume that the maximum TSV count for each die in Set S is also given. Each s_i is a triplet of (x_i, y_i, z_i), where z_i is the die index of s_i, and x_i and y_i are the X and Y coordinates of s_i. Let $stack(S)$ denote the number of dies that the sinks in Set S are located. In each recursive partitioning, we divide Set S into two Subsets S_1 and S_2 based on the following two cases:

- *Z-cut*: if the TSV bound is one, the given Sink Set S is partitioned such that the sinks from the same die belong to the same subset. The connection between S_1 and S_2 needs one TSV between adjacent dies. Note that the 3D-MMM algorithm is a bi-partitioning process. If the sinks in Set S belong to more-than-two dies (i.e., $stack(S) > 2$), we need $stack(S) - 1$ iterations of Z-direction partitions to split the sink set into subsets so that the sinks belonging to the same die are in the same subset. Furthermore, the order of the Z-cut also depends on the source-die index.

- *X/Y-cut*: if the TSV bound is larger than one or the sinks in Set S belong to the same die, Set S is partitioned geometrically by a horizontal line (X-cut or Y-cut) and Z-dimension is ignored. If the subsets contain sinks from different dies, we potentially need multiple TSVs to connect those sinks.

At the end of each partitioning, we propagate the TSV bound to the new subsets.

The 3D abstract tree generation using the 3D-MMM algorithm is shown in Figure 4. The recursive method takes as inputs a set of 3D clock sinks and a TSV bound. If the size of the given sink set (i.e., $|S|$) is one, then we reach the bottom level of the abstract tree (Lines 3-4). If the TSV bound is one, Z-cut is applied to partition Sink Set S into two Subsets S_1 and S_2 (Lines 6-7).

```
3D Abstract Tree Generation (3D-MMM)
Input: clock sinks in 3D and a TSV bound
Output: a rooted 3D abstract tree

1: AbsTreeGen3D(SinkSet $S$, bound $B$)
2:   $S_1$ and $S_2$ = subsets of $S$;
3:   if ($|S| = 1$) then
4:      return $root(S)$;
5:   else if (B = 1 and $stack(S) > 1$) then
6:      Z-cut($S$, $S_1$, $S_2$);
7:      $B_1 = B_2 = 1$;
8:   else
9:      Geometrically divide $S$ into $S_1$, $S_2$;
10:     Find $B_1$, $B_2$ such that $B_1 + B_2 = B$;
11:  $root(S_1)$ = AbsTreeGen3D($S_1$, $B_1$);
12:  $root(S_2)$ = AbsTreeGen3D($S_2$, $B_2$);
13:  $leftChild(root(S)) = root(S_1)$;
14:  $rightChild(root(S)) = root(S_2)$;
15:  return $root(S)$;
```

Figure 4: Pseudo code of the 3D-MMM algorithm.

As previously discussed, once the TSV bound is one, our 3D-MMM algorithm performs $stack(S) - 1$ times of Z-direction partitions. To guarantee that only one TSV is used between adjacent dies, the order of die-wise Z-cut depends on the source-die index and the die indices in Sink Set S. The detailed Z-cut procedure are shown in Figure 5. If the above conditions are not satisfied, Set S is partitioned geometrically by a horizontal line (X-cut or Y-cut), so called X/Y-cut (Line 9). And the Z-dimension of each sink is ignored. The cut line is drawn at the median of the X or Y coordinates of the sinks. The TSV bound is divided for the two subsets (Line 10).

The bound for each subset is calculated by estimating the number of TSVs required by each subset and dividing the given Bound B according to the ratio of the estimated TSVs. For each subset, we assume the minimum sink size in each die as the estimated TSV count. This procedure is called recursively for each of Subsets S_1 and S_2 with different TSV bounds (Lines 11-12). The roots of the subtrees are connected by the root of the higher-level tree (Lines 13-15). The complexity of the algorithm is $O(n \cdot logn)$, where n is the number of nodes.

```
**Z-cut(SinkSet $S$, Subset $S_T$, Subset $S_B$)**
Input: Sink set $S = \{s_1, \cdots, s_k\}$, source die index $Z_s$
Output: Subsets $S_T$ and $S_B$

$Z_{\min} = min(z_1, .., z_i, .., z_k)$, $s_i = (x_i, y_i, z_i) \in S$
$Z_{\max} = max(z_1, .., z_i, .., z_k)$, $s_i = (x_i, y_i, z_i) \in S$
if ($Z_s \le Z_{min}$) then
    $S_T = \{s_1, .., s_i, .., s_{k1}\}$, $z_i \in [Z_{\min} + 1, Z_{\max}]$
    $S_B = \{s_{k1+1}, .., s_j, .., s_k\}$, $z_j = Z_{\min}$
else if ($Z_s \ge Z_{\max}$) then
    $S_T = \{s_1, .., s_i, .., s_{k1}\}$, $z_i = Z_{\max}$
    $S_B = \{s_{k1+1}, .., s_j, .., s_k\}$, $z_j \in [Z_{\min}, Z_{\max} - 1]$
else
    $S_T = \{s_1, .., s_i, .., s_{k1}\}$, $z_i = Z_s$
    $S_B = \{s_{k1+1}, .., s_j, .., s_k\}$, $z_j \ne Z_s$
```

Figure 5: Pseudo code of the Z-cut procedure, which corresponds to Line 6 in the 3D-MMM algorithm in Figure 4.

Corresponding to the n-die stack clock sinks, the 3D abstract tree is an n-colored binary tree, where each node (i.e., sinks, internal nodes, and the root) is assigned a color to represent the die the node belongs to. The dies are numbered bottom up from 1 to n. Let $c(p)$ be the color index of Node p, where $c(p) \in \{1, 2, .., n\}$. For example, $c(p) = 1$ means that Node p is located in die-1. Let $c(src)$ denote the source-die index. In the top-down 3D abstract tree generation, we color the nodes corresponding to the sink sets. Considering Node p with Sink Set S, let $Z_{\max}$ and $Z_{\min}$ be the maximum and minimum die indices in Set S. The color of p is determined as follows:

$$c(p) = \begin{cases} c(src), \text{ if } p \text{ is the root;} \\ Z_{\min}, \text{ else if } Z_{\min} > c(src); \\ Z_{\max}, \text{ else if } Z_{\max} < c(src); \\ c(src), \text{ otherwise.} \end{cases} \tag{1}$$

Considering Edge e with two terminal Nodes n_1 and n_2, the following statements are true: (1) If $c(n_1) = c(n_2)$, Edge e will be routed in the same die as Nodes n_1 and n_2 and (2) if $c(n_1) \neq c(n_2)$, then $|c(n_1) - c(n_2)|$ TSVs will be inserted along Edge e. An illustration is shown in Figure 6, where 3D abstract trees for Sink Set $\{a, b, c\}$ are generated after applying Z-cut twice.

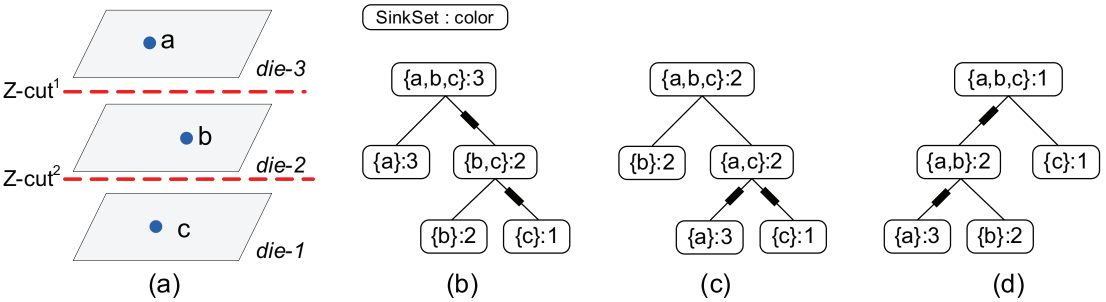


Figure 6: Three-colored 3D abstract trees after applying Z-cut twice on the three-die-stacked Sink Set $\{a, b, c\}$, if the clock source is located in (b) die-3, (c) die-2, and (d) die-1. Each node in the abstract tree contains the corresponding sink set and a color index.

In Figures 6(b), (c), and (d), the clock source is located in die-3, die-2, and die-1, respectively. Each node in the abstract tree contains the sink set and the color information. The abstract tree in Figure 6(b) is obtained by Z-cut^1 first and then Z-cut^2. Whereas, the sequence in Figure 6(d) is Z-cut^2 first and then Z-cut^1. In addition, the abstract tree in Figure 6(c) is generated by first extracting the sinks of the clock-source die and then applying a Z-cut. The primary goal of using different Z-cut sequences is to guarantee that only one TSV is used between adjacent dies after $stack(S) - 1$ Z-cuts.

2.2.3 Slew-Aware Buffering and Embedding

The second step of the 3D clock tree synthesis is the slew-aware buffering and embedding. Given a 3D abstract tree, the goal is to determine the exact geometric locations of all the nodes, TSVs, and buffers. The following requirements are satisfied: (1) The wirelength of the embedded-and-buffered clock tree is minimized; (2) the load capacitance of each buffer does not exceed the pre-defined maximum value (CMAX); and (3) clock skew is zero under the Elmore delay model. We develop the slew-aware deferred-merge buffering and embedding (sDMBE) algorithm to geometrically embed and route the abstract tree.

The sDMBE algorithm consists of two steps and is based on the deferred-merge embedding (DME) algorithm [22]: The first phase in the sDMBE algorithm is to determine the merging types and to construct the merging segments for each pair of subsets in a bottom-up traversal. Different from the existing 2D synthesis [34, 35, 37], which focused on slew-aware buffer insertion after clock routing, the sDMBE method performs buffer insertion in the bottom-up procedure. The goal of slew-aware buffering is to locate buffers while merging subsets so that the load capacitances of buffers are within the given bound (CMAX). The impact of CMAX on the 3D clock slew is discussed in Section 2.4.5. Merging segments are obtained based on the merging distances, which are computed under the zero-skew equations in the Elmore delay model and the wirelength minimization goals. The second phase of the sDMBE algorithm is to decide the exact locations of internal nodes, buffers, and TSVs in a top-down fashion and to determine the routing topology of the overall clock nets. The complexity of our approach is $O(n)$.

Two samples of merging segments for unbuffered and buffered 3D clock trees are shown in Figure 7. When merging Child Nodes u and v to the Parent Node p, the sDMBE algorithm first decides the merging type based on the given 3D abstract tree and the CMAX constraint. Corresponding to the merging type among clock wires, buffers, and TSVs, we obtain the merging distances of Nodes p and u and Nodes p and v in Figure 7(a) and the merging distances between Node p and Buffer b, Buffer b and Node u, and Nodes p and v in Figure 7(b).

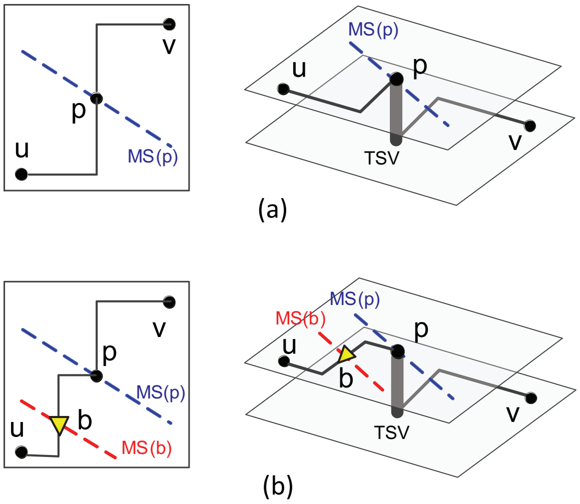
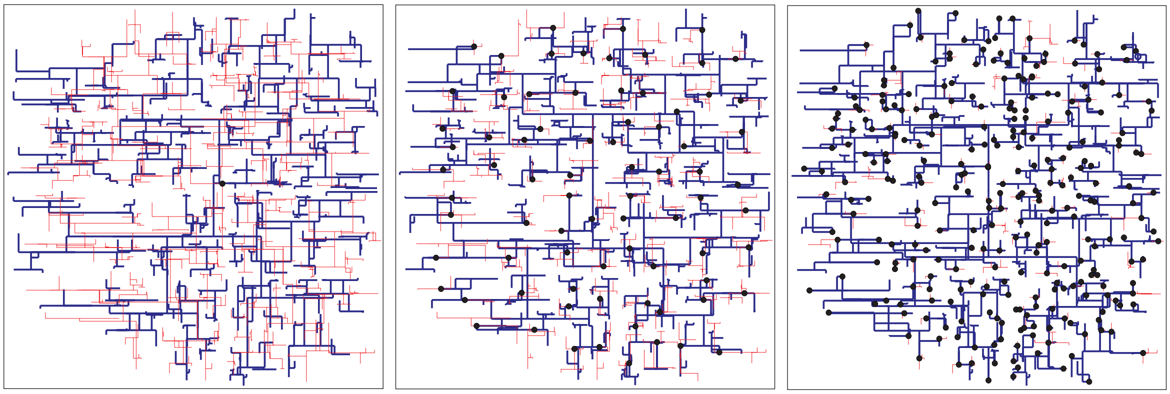


Figure 7: Samples of 3D merging segments for (a) an unbuffered tree and (b) a buffered tree.

2.3 Extension of 3D-MMM Algorithm

Figure 8 provides a demonstration that higher usage of TSVs leads to shorter wirelength than fewer TSVs. This raises an important question: *what is the optimal number of TSVs for a 3D clock tree that leads to the minimum possible power consumption?* One obvious way to answer this question is by trying *all* possible TSV counts and choosing the best power result (an exhaustive search). This method, however, is very time consuming and requires prohibitive runtime. Thus, our goal is to find this TSV count that leads to the minimum (or close-to-minimum) power result in much shorter runtime. This calls for careful attention to the impact of the TSV count not only on the overall wirelength but also the total number of buffers and total TSV capacitance, as these factors equally affect the overall power consumption.



#TSVs = 1, WL = 775 mm

#TSVs = 78, WL = 676 mm

#TSVs = 283, WL = 589 mm

Figure 8: 3D clock trees for the two-die stack r_3 with varying TSV bounds. The black dots are the TSV location candidates. And the bold and thin lines illustrate the clock nets in die-1 and die-2, respectively.

We develop a new low-power 3D clock tree synthesis method, named 3D-MMM-ext, by extending our 3D-MMM algorithm presented in Section 2.2.2. The goal of the 3D-MMM-ext is to construct a low-power clock network by wisely assigning clock TSVs in the 3D abstract tree generation. In each top-down partition, let S be the current sink set. Let $Z(S)$ denote the vertical distance Set S spans, which can be expressed as

$$Z(S) = Z_{\max} - Z_{\min}, \tag{2}$$

where $Z_{\max}$ and $Z_{\min}$ are the maximum and minimum die indices of the sinks within Set S.

Note that $Z(S)$ also indicates the minimum number of TSVs required by the clock network connecting all the sinks in S. Different from the 3D-MMM algorithm, which decides the cut direction (Z-cut or X/Y-cut) based on the TSV bound (Lines 5 and 8 in Figure 4), the key technique of the 3D-MMM-ext is to determine the cutting orientation of the current iteration (i.e., Z-cut or X/Y-cut) by looking ahead to the next cutting iteration, while estimating and comparing the costs of the following two cases:

- Case-1: apply Z-cut at the current iteration and then apply X/Y-cut on each die once in the following iterations;

- Case-2: apply X/Y-cut at the current iteration and postpone Z-cut to the next iteration.

Note that for the n-die stack case, Z-cut means applying die-wise partitions in multiple iterations until the sinks having the same die index are partitioned into the same subset. In Case-1 style partition, Sink Set S has $stack(S) - 1$ times Z-cuts and $stack(S)$ times X/Y-cuts. S in Case-2 has one X/Y-cut and $2 \times (stack(S) - 1)$ Z-cuts. Let S_i^z and S_i^{xy} represent the subsets after case-1 and case-2 style partitions, respectively. The sinks within Set S_i^z (or S_i^{xy}) are in the same die.

An example is depicted in Figure 9, which determines the current cut direction using the 3D-MMM-ext on Sink Set S. Case-1 style partition is shown in Figure 9(a), where Z-cut is applied in the current iteration and then X/Y-cut^1 and X/Y-cut^2 are applied on die-1 and die-2, respectively. Case-2 partition result is illustrated in Figure 9(b). We also show a part of the 3D abstract tree corresponding to Case-1 and Case-2 partitions, respectively. We have the following relation:

$$S = \bigcup_{i=1}^{4} S_i^z = \bigcup_{i=1}^{4} S_i^{xy}. \tag{3}$$

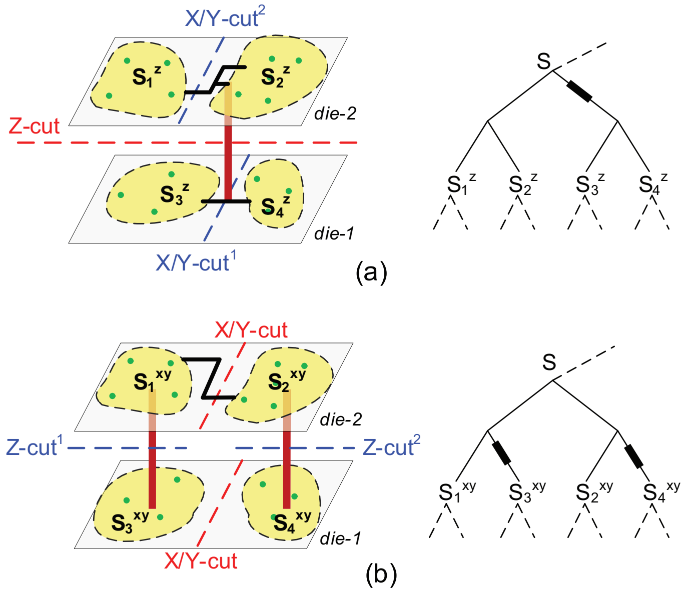


Figure 9: The 3D-MMM-ext algorithm performed on a two-die stack with Sink Set S. We show the 3D abstract trees, cut orders, and the subsets from Case-1 and Case-2 style partitions. (a) Case-1, where we apply Z-cut at the current iteration, and then X/Y-cut^1 and X/Y-cut^2 in die-1 and die-2, respectively. (b) Case-2, where we apply X/Y-cut at the current iteration, and then Z-cut^1 and Z-cut^2. P_z and P_{xy} are the cost of merging S_i^z and S_i^{xy} in (a) and in (b), respectively.

By comparing the cost of Case-1 (P_z) and the cost of Case-2 (P_{xy}), the cut direction of the current iteration is determined as follows:

$$\text{Current Cut} = \begin{cases} \text{X/Y-cut} \text{ , if } P_z > P_{xy}; \\ \\ \text{Z-cut} \text{ , otherwise.} \end{cases} \tag{4}$$

This equation presents that if selecting Z-cut in the current iteration helps reduce power, we choose Z-cut; otherwise, we choose X/Y-cut. Cost P_z is defined as follows:

$$P_z = \sum_{i \in cond1} P(S_i^z) + \sum_{j,k \in cond2} P(S_j^z, S_k^z). \tag{5}$$

Similarly,

$$P_{xy} = \sum_{i \in cond1} P(S_i^{xy}) + \sum_{j,k \in cond2} P(S_j^{xy}, S_k^{xy}). \tag{6}$$

Let S_i represent either S_i^{xy} or S_i^z. The first item $P(S_i)$ in the cost function is the cost of Subset S_i, where $cond1$ covers the final subsets after the look-ahead partitions. The second item $P(S_j, S_k)$ in the cost function is the cost of connecting Subsets S_j and S_k. $P(S_j, S_k)$

mainly comes from TSVs, global wires, and buffers. Therefore, *cond*2 covers all pairs of subtrees in the 3D abstract tree, where we merge those final subsets to their parent Sink Set S in the bottom-up traversal.

Considering the two-die stack examples in Figure 9, P_z and P_{xy} can be expressed as follows:

$$P_z = \sum_{i=1}^{4} P(S_i^z) + P(S_1^z, S_2^z) + P(S_3^z, S_4^z) + P(S_1^z \cup S_2^z, S_3^z \cup S_4^z), \tag{7}$$

$$P_{xy} = \sum_{i=1}^{4} P(S_i^{xy}) + P(S_1^{xy}, S_3^{xy}) + P(S_2^{xy}, S_4^{xy}) + P(S_1^{xy} \cup S_3^{xy}, S_2^{xy} \cup S_4^{xy}). \tag{8}$$

To estimate the cost for each sink set, we use the half-parameter wirelength model for $P(S_i^z)$ and $P(S_i^{xy})$. Then, $P(S_j, S_k)$ is estimated according to the following two conditions.

- If no TSV is required to connect S_j and S_k,

  $$P(S_j, S_k) \approx \mathrm{CD}(S_j, S_k), \tag{9}$$

  where $\mathrm{CD}(S_j, S_k)$ is the distance between the centers of Subsets S_j and S_k. In Figure 9, $P(S_1^z, S_2^z)$, $P(S_3^z, S_4^z)$, and $P(S_1^{xy} \cup S_3^{xy}, S_2^{xy} \cup S_4^{xy})$ belong to this case.

- If TSVs are needed to provide interdie connection between S_j and S_k,

  $$P(S_j, S_k) \approx \mathrm{CD}(S_j, S_k) + \alpha \times C_{\mathrm{TSV}}/c, \tag{10}$$

  where C_{TSV} is the TSV capacitance, c is the unit-length capacitance of the clock line, and α is an estimator representing the cost of TSV insertion. The following empirical equation is used to calculate α as

  $$\alpha = (2 \times |Z(S_j) - Z(S_k)| + 3) \times \beta, \tag{11}$$

  where $\beta = 0.05$, 0.05, and 0.1 if the TSV capacitance is 15 fF, 50 fF, and 100 fF, respectively. In Figure 9, $P(S_1^z \cup S_2^z, S_3^z \cup S_4^z)$, $P(S_1^{xy}, S_3^{xy})$, and $P(S_2^{xy}, S_4^{xy})$ belong to this case.

2.4 Simulations and Discussions

We first examine a two-die stack to investigate the impact of the TSV count and TSV parasitics on clock power consumption. Next, we show the efficiency of the 3D-MMM-ext algorithm in finding the optimal number of TSVs to be used for minimum power consumption. We then present the results of our clock slew control method. Lastly, we show the impact of scaling the supply voltage on 3D clock power consumption. We validate our claims with SPICE simulation results.

2.4.1 Simulation Settings

We construct zero-Elmore-skew 3D clock networks by using the proposed 3D clock tree synthesis methods. We then extract the netlist of the entire 3D clock network for SPICE simulation. After the simulation, we obtain highly accurate power consumption and timing information of the entire clock network. Note that our 3D clock tree has zero skew under the Elmore delay model, but may have nonzero clock skew from SPICE simulation. Thus, we constrain the SPICE clock skew to be less than 3 % of the clock period at a frequency of 1 GHz. The slew is constrained within 10 % of the clock period. Clock power mainly comes from the switching capacitance of the interconnect, sink nodes, TSVs, and clock buffers.

The technical parameters are based on the 45 nm Predictive Technology Model [82]: per unit-length wire resistance is 0.1 Ω/um, and per unit-length wire capacitance is 0.2 fF/um. The buffer parameters are: driving resistance is 122 Ω, input capacitance is 24 fF, and intrinsic delay is 17 ps. The TSV resistance is 35 $m\Omega$. In order to study the impact of the TSV RC parasitics on the 3D clock network, we vary the linear oxide thickness and choose three typical TSV capacitance values (i.e., 15 fF, 50 fF, 100 fF). The supply voltage is set to 1.2 V unless otherwise specified. The maximum load capacitance of each clock buffer, denoted CMAX, is set to 300 fF for slew control unless otherwise specified.

Our analysis focuses on two-die and six-die 3D clock networks. In the six-die case, the clock source is located in the middle die (die-3) as suggested in [77], unless otherwise specified. As a result, die-3 in a six-die clock network contains a complete tree. The IBM benchmarks r_1 to r_5 [83] are used. Since r_1 to r_5 are originally designed for 2D ICs, we randomly distribute the sinks into two or six dies. We then scale the footprint area by $\sqrt{N}$ to reflect the area reduction in the 3D design.

2.4.2 Impact of TSV Count and Parasitic Capacitance

To investigate the impact of the TSVs on clock power consumption, we use a two-die stack implementation of the biggest benchmark r_5, which has 3101 sink nodes with input capacitances varying from 30 fF to 80 fF. Three clock power trend curves are depicted in Figure 10, where the TSV capacitance (C_{TSV}) varies from 15 fF, 50 fF, to 100 fF. On the x-axis, we show the total number of TSVs used in each entire 3D clock tree, which is obtained by imposing a different TSV bound. Our baseline 3D clock network contains only one TSV between adjacent dies.

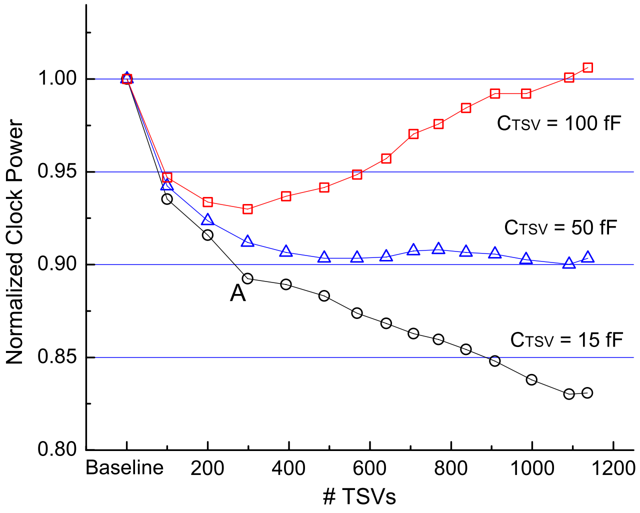


Figure 10: Impact of the TSV capacitance and count on clock power for the two-die $r5$. The TSV capacitance (C_{TSV}) is set to 15 fF, 50 fF, and 100 fF. Our baseline is the clock tree that uses one TSV between adjacent dies. For each C_{TSV}, we show the 3D-MMM results by sweeping the TSV count. We also highlight the 3D-MMM-ext results for each C_{TSV}, which are marked as stars near to the trends.

The clock power is affected by both the TSV count and the TSV capacitance as shown in Figure 10. First, using 15 fF TSVs in the clock network construction, the clock power decreases significantly when more TSVs are used. We are able to obtain a low-power clock network design by relaxing the TSV bound. We can achieve up to 17.0 % power reduction compared with the single-TSV case. The power savings mostly comes from wirelength reduction, because the clock wire capacitance significantly affects the overall power consumed by the clock network. When more TSVs are used, the number of local trees in the non-source dies increases, while their size decreases. This phenomenon means that the multiple-TSV case encourages local clock distribution in 3D designs while reducing the overall wirelength.

Second, if the TSV has a large capacitance (e.g., 50 fF, 100 fF), the contribution of the TSV capacitance to the overall power consumption is non-negligible. As a result, when the TSV count increases, the overall clock power reduction becomes slower. Particularly, if the TSV capacitance is 100 fF, clock power does not decrease when the TSV count exceeds a certain amount and eventually starts increasing. In this case, the clock power from the TSV capacitance increases faster than the power decreases from wirelength reduction.

From this trend study, we conclude that given a TSV parasitic capacitance, there exists an optimum number of TSVs that results in the minimum 3D clock power. This trend in turn allows us to choose the right TSV bound for a given power budget. If a power savings of 10 % is required for using the 15 fF TSVs, the TSV bound of 300 can be used based on Point A in Figure 10.

2.4.3 Exhaustive Search Results

A straightforward way to find the "min-power TSV count", i.e., the number of TSVs used in a 3D clock tree that leads to the minimum overall clock power consumption, is to exhaustively sweep the TSV bound from 1 to infinity[2], constructing and simulating the entire 3D clock network corresponding to each TSV bound. By plotting the TSV count vs. power trend curve, we are then able to find the optimum solution. A clock power trend is depicted in Figure 11, where 1137 3D clock trees are generated and simulated for the two-die stack r_5. We assume the TSV parasitic capacitance is 100 fF.

We observe that the lowest power comes from the clock network that uses 250 TSVs,

[2] Note that the TSV bound of infinity means that we do not impose any restriction on the maximum number of TSVs used in each die. This usually results in a high usage of TSVs that mainly targets at wirelength minimization.

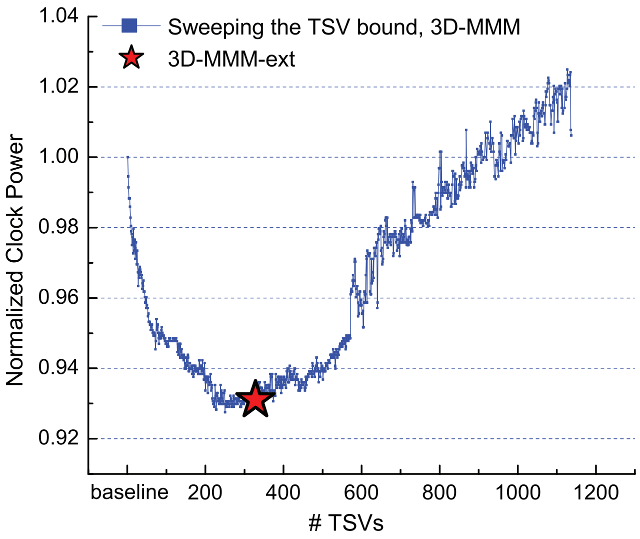


Figure 11: Clock power trends for the two-die stack r_5 based on the exhaustive search within the TSV count range $[1, 1137]$. The TSV capacitance is 100 fF. We also plot the 3D-MMM-ext algorithm result. The exhaustive search covers 1137 simulations on various clock trees. The runtime for each simulation is around 200 seconds.

with 1.190 W clock power and $2,004,250$ µm wirelength. In addition, we observe that the exhaustive search result agrees with the TSV count vs. power trend we presented in the previous section, although power fluctuates locally in a small range of the TSV count. If the TSV count exceeds 600, the clock power is much more sensitive to the TSV count increase. Using one more TSV may lead to the clock power increasing or decreasing by 1 %. This phenomenon is because, when using a large amount of TSVs, the clock network has a large number of smaller local trees, where the TSV capacitance itself is comparable to or even larger than that of a single local clock tree. As a result, using a few more TSVs leads to a large fluctuation in clock power.

The proposed exhaustive search method does allow us to find the min-power TSV count, but it is too costly in terms of runtime. The smaller step size we use for the TSV count in the search, the lower power of a 3D clock network we find, but more simulations as well as runtime are required. Note that the typical SPICE simulation time of a two-die r_5 clock network is around 200 seconds. Repeating this 1137 times is prohibitive.

2.4.4 3D-MMM-ext Algorithm Results

The comparisons between using a single TSV and using multiple TSVs (obtained with 3D-MMM-ext algorithm) cases are summarized in Tables 1 and 2, where two-die and six-die benchmark designs are implemented.

Table 1: Comparison of wirelength (um), power (mW), TSV count(#TSVs), buffer count (#Bufs), simulation runtime (s), and skew (ps) between using single TSV and using multiple TSVs (3D-MMM-ext) for the two-die stacks. The TSV capacitance is 15 fF, 50 fF, and 100 fF.

C_{TSV}		Single TSV					Multiple TSVs (3D-MMM-ext)							
						Run						Run	Red.(%)	
	ckt	WL	#Bufs	Power	Skew	time	#TSVs	WL	#Bufs	Power	Skew	time	WL	Power
15 fF	r_1	291421	327	0.149	10.5	17.6	93	221443	282	0.125	9.3	16.8	24.0	16.1
	r_2	602484	706	0.314	15.4	43.2	211	445647	588	0.255	14.2	32.5	26.0	18.8
	r_3	775194	930	0.410	17.4	55.2	297	583274	779	0.342	13.5	50.5	24.8	16.6
	r_4	1586630	1990	0.855	18.2	122.8	660	1165529	1594	0.698	16.8	107.1	26.5	18.4
	r_5	2341420	2897	1.283	17.0	188.0	1096	1737100	2509	1.065	19.8	187.7	25.8	17.0
50 fF	r_1	291498	327	0.149	12.4	18.1	85	221719	293	0.130	11.9	17.6	23.9	12.8
	r_2	602485	706	0.314	15.2	38.4	205	448195	618	0.271	13.6	36.5	25.6	13.7
	r_3	775056	930	0.410	17.2	53.2	288	589654	845	0.366	15.7	48.1	23.9	10.7
	r_4	1586880	1991	0.855	14.8	121.5	639	1165253	1727	0.745	15.0	114.6	26.6	12.9
	r_5	2341360	2897	1.283	16.8	220.1	1020	1749543	2684	1.151	17.8	186.3	25.3	10.3
100 fF	r_1	291421	328	0.149	9.9	17.5	45	238242	303	0.137	12.6	16.0	18.2	8.1
	r_2	601929	707	0.313	13.5	40.0	87	492966	661	0.287	13.0	33.5	18.1	8.3
	r_3	775029	930	0.410	17.3	54.2	112	645062	897	0.383	13.4	55.1	16.8	6.6
	r_4	1586630	1992	0.855	15.7	131.3	247	1286784	1891	0.787	18.2	125.2	18.9	8.0
	r_5	2341460	2897	1.283	17.1	187.6	328	1953453	2798	1.194	19.0	179.8	16.6	6.9

Table 2: Comparison of wirelength (um), power (mW), TSV count (#TSVs), buffer count (#Bufs), simulation runtime (s), and skew (ps) between using single TSV and using multiple TSVs (3D-MMM-ext) for the six-die stacks. The TSV capacitance is 15 fF, 50 fF, and 100 fF.

		Single TSV					Multiple TSVs (3D-MMM-ext, src in die-3)							
TSV						Run						Run	Red. (%)	
Cap	ckt	WL	#Bufs	Power	Skew	time	#TSVs	WL	#Bufs	Power	Skew	time	WL	Power
15 fF	r_1	272109	332	0.144	19.4	19.0	297	138223	214	0.092	12.8	10.5	49.2	36.1
	r_2	566944	684	0.298	16.1	45.0	668	280901	445	0.191	18.2	29.7	50.5	35.9
	r_3	717479	887	0.388	15.0	57.0	965	376634	626	0.264	17.1	45.8	47.5	32.0
	r_4	1496180	1870	0.816	18.5	119.8	2195	752370	1316	0.551	17.6	84.0	49.7	32.5
	r_5	2299220	2935	1.265	19.6	205.3	3497	1133262	2070	0.854	21.4	154.0	50.7	32.5
50 fF	r_1	272849	332	0.144	17.4	17.7	275	143626	257	0.106	18.5	11.5	47.4	26.4
	r_2	567686	684	0.299	15.0	46.6	631	302068	562	0.230	20.3	35.2	46.8	23.1
	r_3	719610	891	0.389	14.3	66.1	918	403235	775	0.316	18.5	50.2	44.0	18.8
	r_4	1493990	1870	0.815	15.0	123.0	2045	810708	1680	0.670	27.0	95.1	45.7	17.8
	r_5	2299590	2935	1.266	19.3	217.8	3270	1250269	2644	1.051	23.4	189.8	45.6	17.0
100 fF	r_1	273951	332	0.145	16.6	16.8	30	234821	309	0.133	29.0	17.1	14.3	8.3
	r_2	566803	685	0.298	11.1	45.1	80	468805	638	0.271	28.9	41.2	17.3	9.1
	r_3	720705	893	0.390	14.2	61.6	75	651298	873	0.374	23.1	60.3	9.6	4.1
	r_4	1497240	1873	0.817	14.0	126.5	115	1333034	1804	0.769	23.8	118.8	11.0	5.9
	r_5	2300620	2935	1.266	19.2	183.6	180	2014167	2780	1.179	28.3	186.7	12.5	6.9

First, the 3D-MMM-ext is able to find the low-power 3D clock trees. For the two-die stacks in Table 1, the 3D-MMM-ext reduces the clock power by around 16.1 % to 18.8 %, 10.3 % to 13.7 %, and 6.6 % to 8.3 % as compared with the single-TSV cases and achieves wirelength savings around 24.0 % to 26.5 %, 23.9 % to 26.6 %, and 16.6 % to 18.9 %, when the TSV capacitance is 15 fF, 50 fF, and 100 fF, respectively. In the case of six-die stacks shown in Table 2, our 3D-MMM-ext reduces power by up to 36.1 %, 26.4 %, and 9.1 %, and reduces wirelength by up to 50.7 %, 47.4 %, and 17.3 %.

In most cases, the simulated clock skew is less than 20 ps, which is less than the 30 ps constraint. In the case of the six-die 3D stack of r_5, the spatial distribution of the propagation delay is plotted in Figure 12, where the die contains the clock source.

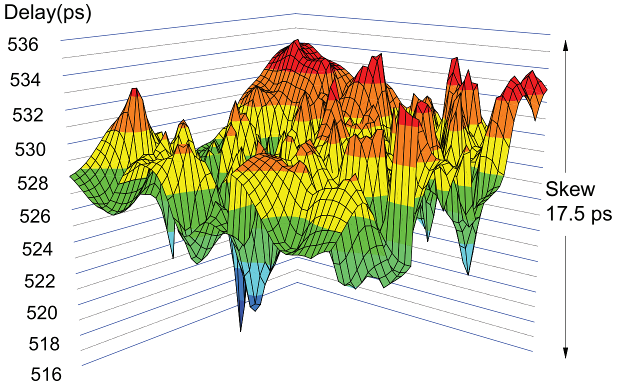


Figure 12: Spatial distribution of propagation delay (ps) and clock skew (ps) of the clock source die for the six-die stack r_5. The TSV count is 3497.

The TSV count is 3497. We observe that the clock skew among the six dies varies within $[17.5\text{ ps}, 21.4\text{ ps}]$. The skew of the entire 3D clock network is 21.4 ps. Referring to the TSV RC parasitics and the 300 fF CMAX constraint, the delay along each TSV is in the order of 0.01 ps. Compared with the $>$ 500 ps src-to-sink delay, the TSV contributes a negligible portion of delay to the entire src-to-sink delay. Note that our 3D clock tree synthesis algorithm builds a zero-skew tree under the Elmore delay model, which in practice shows discrepancy between SPICE simulation results.

2.4.5 Low-Slew 3D Clock Routing

The TSV count can also affect the clock slew distribution. The slew distributions of the six-die 3D clock tree for r_5 among all sinks are depicted in Figure 13. The clock slew constraint is set to 100 ps, which is 10 % of the clock period. The slew distribution of the single-TSV clock tree is shown in Figure 13(a), whereas Figure 13(b) is the slew distribution of the multiple-TSV clock tree using the 3D-MMM-ext.

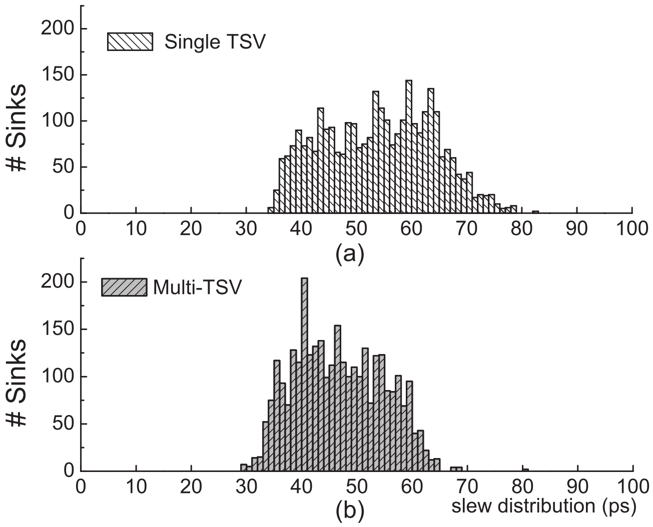


Figure 13: Slew distribution of six-die 3D clock network among all sinks. Slew constraint is set to 10 % of the clock period, and CMAX is 300 fF. (a) Slew distribution in the single-TSV clock tree, (b) in the multiple-TSV clock tree.

In the single-TSV clock tree, slew varies within [34.2 ps, 82.7 ps] with an average slew of 53.9 ps. The slew distribution of the multiple-TSV case is in the range of [29.1 ps, 80.3 ps] with an average slew of 46.8 ps. Compared with the single-TSV case, the multiple-TSV case reduces the maximum slew and average slew by 2.4 ps and 7.1 ps, respectively. The main reason for the improved slew distribution of the multiple-TSV 3D tree is the shorter wirelength, which in turn reduces the capacitive load. Thus, we conclude that multiple TSVs are effective in improving the slew distribution.

The impact of the maximum clock buffer load capacitance (CMAX) on the slew variations (min, average, max) and power consumption in the single-TSV and multiple-TSV clock trees is shown in Figure 14. First, CMAX remains as an efficient means to control the maximum slew in 3D clock network design. Both the single-TSV and multiple-TSV cases have similar trends as CMAX varies from 300 fF to 175 fF: a smaller CMAX reduces the maximum slew but increases the clock power. This phenomenon is because each buffer stage is allowed to drive a smaller capacitance with smaller CMAX, which in turn requires more buffers and thus consumes more power. Second, given a certain CMAX, multiple-TSV clock trees always have reduced maximum slew and less average slew than the single-TSV cases. Third, the multiple-TSV case always consumes less power than the single-TSV case. Therefore, we conclude that the multiple-TSV case achieves both low power and better slew results.

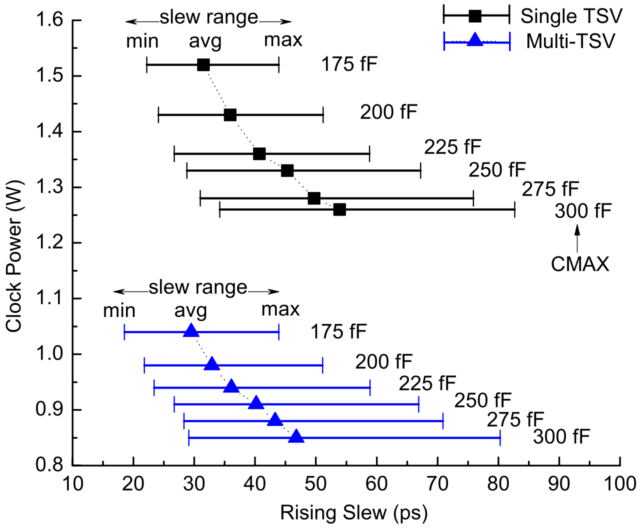


Figure 14: Slew variations and power comparisons between single-TSV and multiple-TSV clock trees. CMAX varies from 175 fF to 300 fF.

2.5 Summary

In this chapter, we explored design optimization techniques for reliable low-power and low-slew 3D clock network design. We thoroughly studied the impact of the TSV count and the TSV capacitance on clock power trends. We observed that using more TSVs helps reduce the wirelength and power consumption and shows better control over clock slew variations. However, in the case of a large TSV parasitic capacitance, clock power could increase if too many TSVs are used. We also observed that a smaller maximum loading capacitance on the clock buffers efficiently lowers the 3D clock slew. Furthermore, we developed a low-power

3D clock tree synthesis algorithm called 3D-MMM-ext. Experimental results show that our 3D-MMM-ext algorithm constructs low-power 3D clock designs that have comparable power and reliability to an exhaustive search approach with a few orders of magnitude shorter runtime.

CHAPTER III

CLOCK NETWORK DESIGN FOR PRE-BOND TESTING OF 3D-STACKED ICS

Three-dimensional system integration has emerged as a key enabling technology to continue the scaling trajectory predicted by Moore's Law for future IC generations. With 3D integration technology, both the average and maximum distance between components can be substantially reduced by placing them on different dies, which translates into significant savings in delay, power, and area. Moreover, it enables the integration of heterogeneous devices, making the entire system more compact and efficient. Nevertheless, the success of 3D stacked ICs is predicated on the final post-bond yield, i.e., minimizing the number of good dies bonded to defective dies. Therefore, each die must be tested prior to the bonding process.

In Chapter 2, we demonstrated that there exists a TSV vs. wirelength (and thus power) tradeoff in 3D clock trees: the more TSVs used in the 3D clock tree, the shorter the total wirelength. This discussion clearly motivates using more TSVs in a 3D clock tree. However, the 3D clock trees containing multiple TSVs have an interesting property: only one die in the stack contains a fully connected 2D clock tree; the other dies contain many small, isolated subtrees. These trees take advantage of TSVs to shorten the total wirelength, but such a design makes pre-bond testing next to impossible because each clock subtree requires its own probe pad. The state-of-the-art testing equipment, e.g., from [84], has more than ± 100ps overall timing accuracy (OTA). This makes it very challenging to use multiple clock probe pads to provide a low-skew clock signal. In addition, the cost of dedicating so many probes to a single signal is significant.

This chapter presents the first work on 3D clock tree synthesis for pre-bond testing. The pre-bond testable clock tree can be used for both pre-bond test and post-bond operation. Two circuit elements are introduced specifically, a *TSV-buffer* and a *redundant tree*, to enable efficient pre-bond testing while minimizing the overall wirelength and clock power. Furthermore, the impact of the parasitic TSV capacitance on pre-bond testable clock trees in terms of wirelength, buffer count, and clock power is discussed. A large TSV capacitance tends to increase the wirelength and the number of buffers required, thus increases the clock power. Compared with the simple pre-bond testability solution of using a single TSV to connect two complete 2D trees, the proposed approach significantly reduces the wirelength and power consumption in both two-die and four-die 3D stacks.

3.1 Problem Formulation

The pre-bond testable 3D clock routing problem is defined as follows: given a set of clock sinks distributed across N dies (where $N > 1$) and a TSV bound, construct a 3D clock tree such that (1) during post-bond operation, the tree connects all the sinks with a minimum-skew clock signal, and (2) during pre-bond test, a single 2D clock tree exists in each die that provides a minimum-skew clock signal to the sinks in that die. The objective is to minimize the wirelength and clock power given the TSV bound and the clock slew bound constraints. The clock sinks may represent flip-flops, clock input pins for IP blocks, or memory blocks. Our pre-bond testable clock routing algorithm can operate under any TSV bound greater than zero, and it constructs a high quality 3D clock tree in terms of clock skew[1], wirelength, power consumption, and clock slew for both pre- and post-bond testing and operations.

3.2 Pre-Bond Testable Clock Routing

3.2.1 Overview

Without loss of generality, we first develop a pre-bond testable clock routing algorithm for a two-die stack. We extend it to the stacks containing more-than-two dies in Section 3.2.5. The input to our algorithm includes the location and capacitance of the sinks in each die (die-0 and die-1), a TSV bound (> 0), and a slew constraint. Die-0 is assumed to contain the clock source. Our algorithm consists of two main steps.

[1] In the pre-bond testable clock routing, our algorithm generates zero-skew clock trees based on the Elmore delay model [21]. To obtain accurate clock-related metrics, we then extract the netlist, and report the SPICE simulation results, including delay, skew, slew, and power consumption.

- **3D tree construction**: we generate a 3D clock tree (*post-3d*) connecting all the sinks in both dies so that (1) the overall 3D tree is zero skew under the Elmore delay model; (2) the total wirelength is minimized; and (3) die-0 contains a fully connected 2D tree (*pre-die-0*) with zero skew. In this case, the 3D tree is used during post-bond test and operation, while the 2D tree in die-0 is used for the pre-bond test of die-0. We utilize so called "TSV-buffers" to ensure that the 2D tree in die-0 maintains zero skew in *both* pre-bond and post-bond configurations.

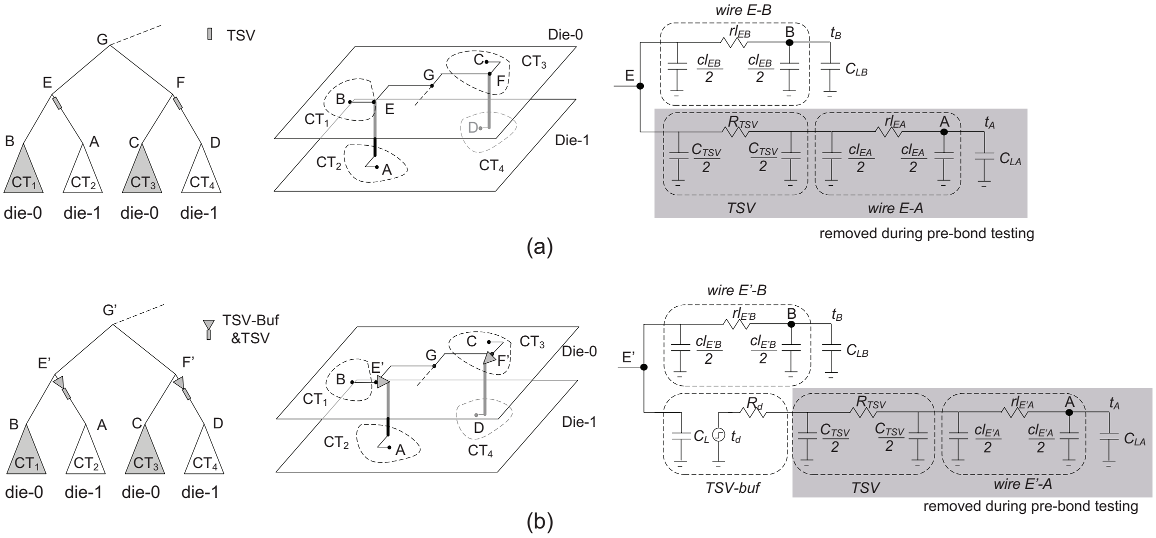
- **Redundant tree routing**: if multiple TSVs are used, the 3D tree construction step generates a 3D tree, where die-1 contains several separate subtrees (*sub-die-1*). In this case, we route a so-called "redundant tree" in die-1 (*red-die-1*) to connect the roots of the subtrees in die-1, and form a single fully connected 2D tree (*pre-die-1*) with (1) an estimated zero skew, and (2) a minimum total wirelength. This 2D tree is used for the pre-bond test of die-1. Transmission gates (*TG*s) are inserted to disconnect the redundant tree for post-bond operation.

3.2.2 TSV-Buffer Insertion

Testing die-0 pre-bond requires a fully connected clock tree in die-0 so that the clock signal is delivered to all die-0 sinks using a single test probe. As mentioned earlier, if multiple TSVs are used, the 3D tree construction step gives a 3D tree, where die-0 contains a single fully-connected tree and die-1 contains a forest of small subtrees. During pre-bond test, the two dies are separated and tested individually. In this case, the 2D tree in die-0 can be used without any additional modification. However, the skew of this tree may no longer be zero because the downstream capacitances of the subtrees in die-1 are not present. This additional skew will either slow down or corrupt the testing process.

To avoid this high-skew situation, we employ our TSV-buffer, simply a buffer inserted right before a TSV. In our test-aware DME (*TaDME*) algorithm, we add a TSV-buffer for each TSV and route the tree accordingly under the zero-skew constraint. In this case, the TSV-buffers are inserted in die-0, where the clock source is located. Since the buffers shield die-0 from the downstream capacitance, die-0 remains zero-skew when tested pre-bond. The outcome of *TaDME* is a zero-skew 3D tree that contains a zero-skew 2D tree in die-0 for pre-bond test.

In what follows, we describe how our *TaDME* algorithm modifies the traditional DME algorithm to construct a zero-skew 3D clock tree in the presence of TSV-buffers. A key step in *TaDME* is the bottom-up recursive tree merging. Given a pair of zero-skew subtrees that must be merged, our goal is to determine the merging segment (the set of potential locations for the merging points) and to connect it to the root nodes of the subtrees so that the new merged tree has zero skew. The traditional merging process as used in the original DME algorithm is illustrated in Figure 15(a), where the merging segment of internal Node E is determined based on the parasitics of the TSVs, wires, downstream capacitances, and internal delays of the two subtrees. In this case, if the right branch (TSV, Edge (E, A), and CT_2) of the overall tree is missing, the delay from E to B will change because of the change in the downstream capacitance at Node E. However, if we use a TSV-buffer as shown in Figure 15(b), the delay from E' to B will not change, even if we remove the right branch. This is because the TSV-buffer hides the downstream capacitance at Node E'.



(a)

(b)

Figure 15: (a) A 3D clock tree built with TSVs, where the separation of die-0 and die-1 skews the tree in die-0. (b) A 3D clock tree built with TSV-buffers, where the separation of die does not skew the die-0 tree.

The following notations are used in Figure 15: r and c denote the unit-length wire resistance and capacitance, respectively; R_d is the output resistance of a buffer; C_L is the input capacitance of a buffer; and t_d is the intrinsic delay of a buffer; R_{TSV} and C_{TSV} are the resistance and capacitance of a TSV. Die-0 contains Subtree CT_1 with Root B and a loading capacitance C_{LB}. The internal delay from B to the sinks of CT_1 is t_B. Similar symbols are used for CT_2. A clock wire of length l is modeled as a π-type circuit with a resistor (rl) and two capacitors ($cl/2$). We also model the TSVs with π-type circuits with resistance R_{TSV} and two capacitances $C_{TSV}/2$. Note that the downstream capacitance at internal Node E' in Figure 15(b) is $cl_{E'B} + C_{LB} + C_L$ *both* before and after the dies are bonded. Thus, TSV-buffers allow us to build a 3D tree for die-0 that has zero skew in both pre-bond and post-bond operations.

In the bottom-up merging process, we require that the delay from E' to sinks in CT_1 (through $B = d_{E',CT_1}$) is equal to the delay to the sinks of CT_2 (through $A = d_{E',CT_2}$). That is,

$$d_{E',CT_1} = d_{E',CT_2}. \tag{12}$$

Referring to the merging structure in Figure 15(b), d_{E',CT_1} and d_{E',CT_2} can be expressed as

$$\begin{aligned} d_{E',CT_1} &= rl_{E'B}(cl_{E'B}/2 + C_{LB}) + t_B, && (13) \\ d_{E',CT_2} &= t_d + R_d(C_{TSV} + cl_{E'A} + C_{LA}) + R_{TSV}(C_{TSV}/2 + cl_{E'A} + C_{LA}) + \\ & \quad rl_{E'A}(cl_{E'A}/2 + C_{LA}) + t_A, && (14) \end{aligned}$$

where t_A is the internal delay from A to sinks of CT_2, and C_{LA} is the downstream capacitance of Node A. If there is no detour, the distances between E' and A ($l_{E'A}$) and between E' and B ($l_{E'B}$) can be expressed as

$$l_{E'B} + l_{E'A} = L, \tag{15}$$

where L is the minimum merging distance between A and B. $l_{E'A}$ and $l_{E'B}$ can be determined by solving Equations (12), (13), (14), and (15).

If $l_{E'A}$ or $l_{E'B}$ is negative, a wire detour is required. For example, when $l_{E'A}$ is negative, $l_{E'B}$ must be longer than L to obtain a zero-skew merging. In this case, $l_{E'A}$ is set to zero, and $l_{E'B}$ is calculated by solving Equations (12), (13), and (14). If the calculated $l_{E'B}$ is too long, we insert a clock buffer along Edge $E'B$. Equation (13) is updated correspondingly.

The decision to avoid a detour with a buffer is made by a cost function that considers the capacitance of clock wires, buffers, and TSVs. We use a wire detour if the cost is less than that of buffer insertion, and satisfies the slew constraint.

3.2.3 Redundant Tree Insertion

The pre-bond test of die-1 requires a fully connected clock tree so that the clock signal is delivered to all the sinks in die-1 from just a single test probe. As mentioned earlier, when multiple TSVs are used for wirelength reduction, the 3D tree construction generates a forest of subtrees in die-1. Therefore, our goal is to combine these subtrees into a single fully connected clock tree with zero clock skew and minimum overall wirelength. We accomplish this by adding a redundant tree that connects the roots of the subtrees while maintaining zero skew. We use this fully connected tree during the pre-bond test of die-1. Note that the redundant tree is not used during post-bond test and operation. We use TGs to disconnect the redundant tree.

The redundant tree routing is done using a conventional algorithm as follows: (1) construct a binary abstract tree in a top-down fashion; (2) insert a TG at each sink node; and (3) embed and buffer the abstract tree under the zero-skew and minimal wirelength goals. A sample flow is illustrated in Figure 16.

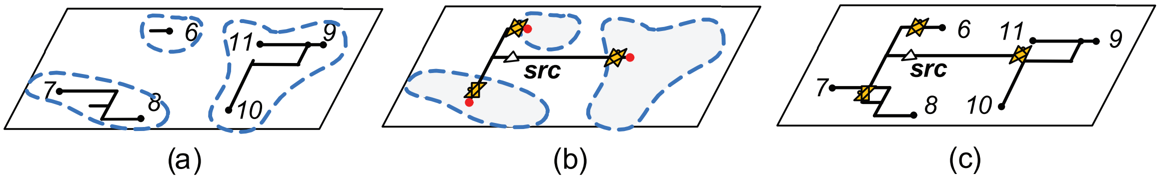


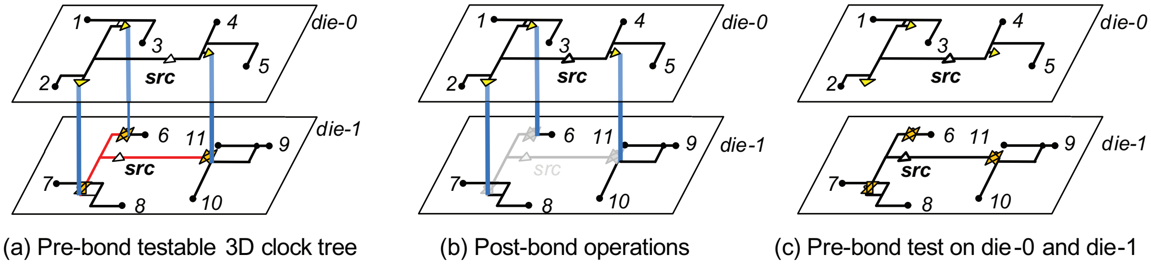
Figure 16: The redundant tree insertion in die-1. (a) Extract sinks from subtrees. (b) Generate a redundant tree and insert transmission gates. (c) The final pre-bond testable clock tree in die-1. The extra control signal that connects the transmission gates is not shown here for simplicity.

Given many subtrees in die-1, we first extract a new set of sinks based on the subtrees as in Figure 16(a). Then, we construct a 2D clock tree for this extracted set as in Figure 16(b). The final pre-bond testable clock tree in die-1 (*pre-die-1*) is illustrated in Figure 16(c), which consists of three subtrees (*sub-die-1*) and one redundant tree (*red-die-1*). Last, we connect the enable input of the TGs using an extra control wire. To minimize the routing overhead, we need to minimize the total wirelength of this control signal. We use the rectilinear minimum spanning tree algorithm (RMST-pack) [85] for this purpose. The cost of this overhead is reported in Section 3.4.3.

3.2.4 Putting It Together

Upon the completion of our algorithm, we obtain fully connected zero-skew 2D clock trees for both die-0 and die-1 as well as a fully connected zero-skew 3D tree for the entire stack. In die-1, we turn on the TGs to connect the redundant tree to the subtrees for pre-bond test. Once the pre-bond testing is complete, we turn off the TGs to disconnect the redundant tree. By doing this, the original zero-skew 3D tree is used for post-bond test and normal operation. We will show in our experimental results that our 3D trees with multiple TSVs, TSV-buffers, and TGs plus the control signal consume significantly less power than a simple single-TSV solution.

The entire design flow is illustrated in Figure 17(a). In post-bond operation, the TGs are turned off and the *pre-die-0* and *sub-die-1* trees are connected with TSVs to form the *post-3d* tree as shown in Figure 17(b). In pre-bond test, the *pre-die-0* tree can be reused with zero skew to test die-0 as shown in Figure 17(c). To test die-1, we turn on the TGs, and the *red-die-1* and *sub-die-1* trees form the zero-skew *pre-die-1* tree, as shown in Figure 17(c).



(a) Pre-bond testable 3D clock tree (b) Post-bond operations (c) Pre-bond test on die-0 and die-1

Figure 17: Example of the post-bond operations and pre-bond test using our 3D clock tree. (a) A pre-bond testable 3D clock tree; (b) a *post-3d* in post-bond operation with TGs turned off; (c) *pre-die-0* and *pre-die-1* in pre-bond test with TGs turned on.

3.2.5 Multiple-Die Extension

For a stack with more-than-two dies, we face the same challenges of creating clock trees for pre-bond test. We take a four-die stacked clock tree in Figure 18 as an example. The clock source is located in die-0. If we apply the 3D-MMM algorithm [45], the resulting *post-3d* tree contains the following topology: (1) die-0 has a complete clock tree connecting all the sinks in die-0; (2) the non-source dies (die-1, die-2, and die-3) have each a *sub-die-k* ($k = 1,\ 2,\ 3$), which are connected to the clock source through 10 TSVs.

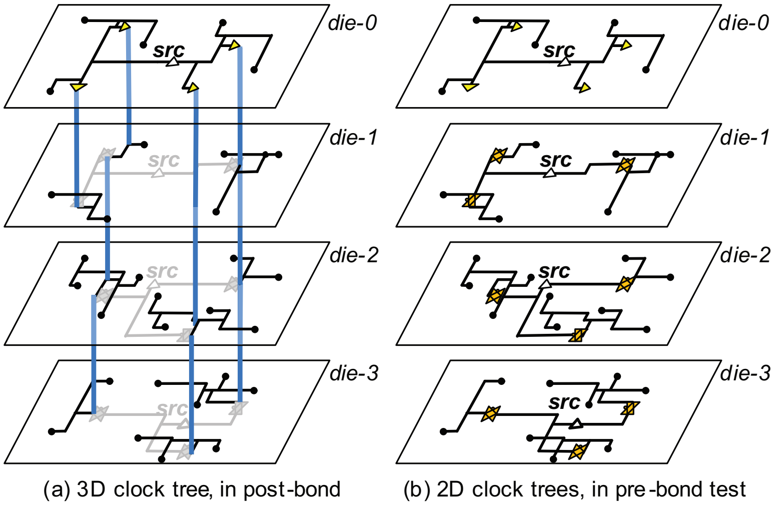

die-0
src
die-1
src
die-2
src
die-3
src
die-0
src
die-1
src
die-2
src
die-3
src
(a) 3D clock tree, in post-bond
(b) 2D clock trees, in pre-bond test

Figure 18: An example of a pre-bond testable clock routing in a four-die stack.

Our pre-bond testable clock routing algorithm for a two-die stack can be easily extended to larger die stacks with an arbitrary clock source location. Our basic 3D tree construction algorithm generates a 3D tree, where die-s (defined as containing the clock source; die-0 in Figure 18) has a single, fully-connected tree, while all the other dies have a forest. During the bottom-up merging process, the TSV-buffer insertion algorithm is extended as follows:

- If a TSV connects die-s and a non-source die-k, where (k ≠ s), we insert a TSV-buffer in die-s;
- If a TSV connects non-adjacent dies and passes through die-s (e.g., connecting die-(s-1) and die-(s+1)), we insert a single TSV-buffer in die-s;
- If a TSV does not connect to or travel through die-s, no TSV-buffer is required.

Once the TSV-buffer insertion, embedding, and buffering are completed, we add redundant trees to the non-source dies. In addition, we insert TGs at the root of each subtree and add a global control signal to connect all the TG enable inputs in each die. This operation allows us to use the redundant trees for pre-bond test (TGs on) and disable them during post-bond test and operations (TGs off). The whole process generates the following items: (1) a single zero-skew 3D clock tree for post-bond test and normal operation; (2) a zero-skew 2D clock tree in each die for pre-bond test; and (3) a global control signal that connects the enable inputs of the TGs in each die. The pre-bond testable and post-bond operational 3D clock tree for a four-die stack is illustrated in Figure 18.

3.3 Buffering for Wirelength and Slew Control

Our pre-bond testable 3D clock routing algorithm inserts two kinds of buffers: clock buffers and TSV-buffers. Clock buffers are mainly used to control delay and skew. These clock buffers are usually inserted close to the clock source and drive large loads to reduce the delay along the clock paths. The TSV-buffers, as discussed in Section 3.2.2, are inserted at every TSV location in the clock source die, so that the clock tree in that die has also zero skew during pre-bond test.

Our observations indicate that the TSV-buffers may unbalance the wirelength during the bottom-up merging process. Considering the example of Subtrees CT_1 and CT_2 in die-0 and die-1, respectively, we must use a TSV-buffer in die-0 to merge these subtrees. As shown in Figure 15(b), the TSV-buffer insertion can increase the delay from E' to CT_2. If the internal delay of CT_2 is already much greater than that of CT_1, adding the TSV-buffer only makes the difference worse. If the difference is too large, wire snaking is required to balance the delays and to achieve a zero-skew merged tree. Thus, the addition of a TSV-buffer has led to a significant clock wirelength overhead in die-0.

To mitigate this overhead, we add extra clock buffers to die-0 to balance the internal delays and eliminate snaking. Specifically, when a TSV-buffer significantly unbalances the delay, we insert an extra clock buffer on the other branch as a counter balance. In Figure 15(b), we add an extra clock buffer along E'-B. We observe that this delay balancing scheme reduces the overall wirelength in die-0. We also observe that few clock buffers are required in this way because such unbalances do not occur frequently.

Clock slew rate control is an important reliability issue for high-speed clocking. If the slew rate is too low – that is, if it takes too long for the clock signal to rise or fall – setup and hold times may be violated. This hold time violation cannot be fixed with a lower clock frequency. Existing work on slew-aware clock tree synthesis relies on buffer insertion [34–37]. Buffers are added along the clock paths so that the output load of each buffer is limited. This bounding condition, denoted as *cmax* in the literature, is shown to be effective in controlling the slew rate. A smaller *cmax* value improves the slew rate, but requires more buffers. Most existing studies insert buffers in a given clock tree as a post-processing step to improve the slew rate under various constraints: buffer area, clock power, etc. This post-synthesis slew-aware buffer insertion must be done carefully to avoid introducing new clock skew, which may constrain the location of the buffers.

Our strategy is to tackle the slew rate issue *during* the construction of the pre-bond testable clock trees by adding buffers to meet the *cmax* constraint. Specifically, we insert clock buffers, together with TSV-buffers, during the bottom-up merging process so that *cmax* is satisfied for both types of buffers. We add clock buffers along the paths from the merging node to the subtree root nodes if the downstream capacitance at the merging node exceeds *cmax*. Depending on the load, we may insert multiple clock buffers to meet the *cmax* requirement.

Several possible scenarios for the clock buffer and TSV-buffer insertion are illustrated in Figure 19. In summary, our clock tree synthesis algorithm uses the following three criteria for buffer insertion during the bottom-up merging process:

- For pre-bond testability, we add a TSV-buffer for every TSV connecting to the clock source die;
- For wirelength reduction, we add a clock buffer to correct unbalances in the delays of two merging subtrees as discussed in the previous section;
- For slew rate control, we add clock buffers, if the downstream capacitance of any buffer exceeds the given limit *cmax*.

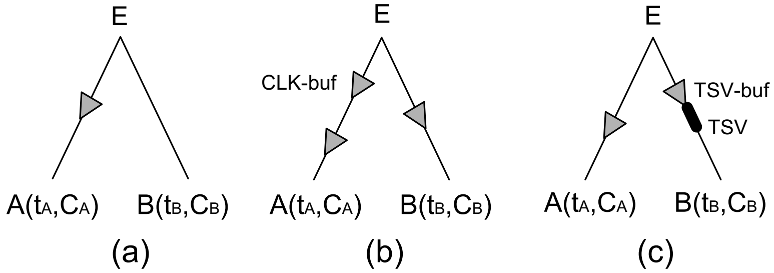


Figure 19: Examples of the clock buffer and TSV-buffer insertion. (a) A clock buffer is inserted to balance the delay of the two branches, where $t_A < t_B$. (b) Multiple clock buffers are inserted if the wires are long and/or the download capacitance is large. (c) A clock buffer is inserted along with a TSV-buffer to balance the delay.

3.4 Experimental Results

We implemented our algorithm using C++/STL on Linux. We use five benchmarks from the IBM suite [83] and four from the ISPD clock network synthesis contest suite [86]. Since these designs are for 2D ICs, we obtain 3D designs by randomly partitioning the clock sinks across the multiple dies and scaling the footprint area by $\sqrt{2}$ and $\sqrt{4}$ for two-die and four-die stacks, respectively.

We use technology parameters from the 45 nm Predictive Technology Model (PTM) [82]; the unit-length wire resistance is 0.1 Ω/μm, and the unit-length wire capacitance is 0.2 fF/μm. The sink capacitance values range from 5 fF to 80 fF. The buffer parameters are R_d = 122 Ω, C_L = 24 fF, and t_d = 17 ps. We use 10 μm × 10 μm via-last TSVs with 20 μm height and 0.1 μm liner oxide thickness. By simulating the TSV structure with Synopsys Raphael [87], we determine the TSV parasitics to be $R_{TSV} = 0.035$ Ω and $C_{TSV} = 15.48$ fF. The clock frequency is set to 1 GHz and the supply voltage (Vdd) to 1.2 V[2]. The maximum load capacitance for each buffer *cmax* is 300 fF for slew rate control.

In SPICE simulation, wire segments and TSVs are represented as π models, and clock buffers and TSV-buffers are represented as inverter pairs. The simulated clock skew and slew tolerances are 3 % and 10 % of the clock period, respectively. We report wirelength in μm, clock power in mW, skew and slew in ps, and capacitance in fF.

[2] Note that our clock trees with single and multiple TSVs are simulated under the same Vdd, and the power savings mainly come from the capacitance reduction. Therefore, the efficiency of our algorithm in low power and pre-bond testability apply on different Vdd (e.g., from 1.2 V to 1.0 V).

3.4.1 TSV-Buffer and TG Model Validation

In pre-bond testable clock routing, we utilize TSV-buffers and TGs to facilitate pre-bond test, and post-bond test and operation. The equivalent circuits are shown in Figure 20, which are used for SPICE validation of the TSV-buffers and TGs. We simulate a post-bond 3D clock tree in a two-die stack and two pre-bond testable 2D clock trees in die-0 and die-1. Node A is the clock source for post-bond operation. Sink C in die-0 and Sink E in die-1 have loading capacitances of C_{LC} and C_{LE}, respectively. Nodes B and D are connected by a TSV-buffer and a TSV. Edge (D, E) is a subtree in die-1 and is connected to F, the clock source for pre-bond test of die-1, via a TG. C_{LC} and C_{LE} are set to 5 fF. Wires (A, B), (B, C), (D, E), and (F, D) all have 500 µm length.

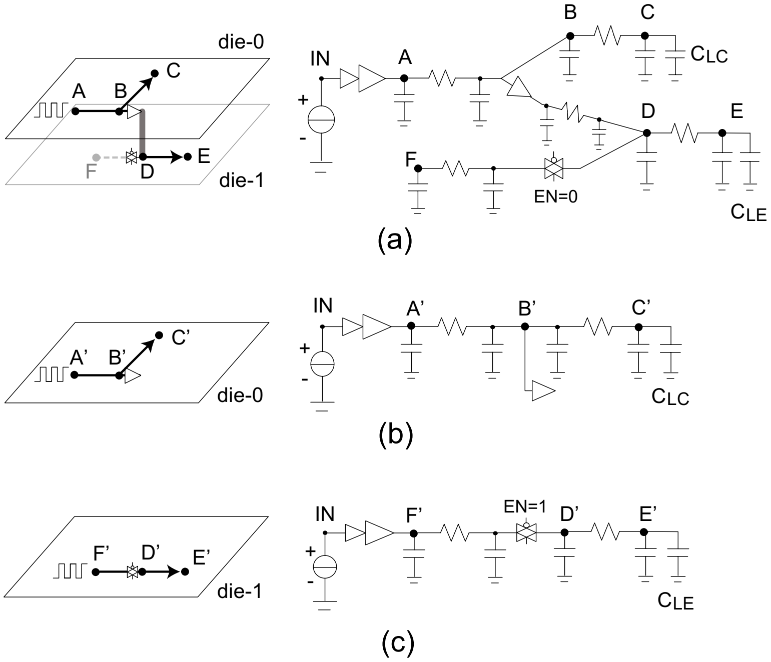


Figure 20: Circuit models for (a) the post-bond 3D clock tree, (b) the pre-bond testable 2D clock tree in die-0, and (c) the pre-bond testable 2D clock tree in die-1.

First, we observe from SPICE simulation that the delay from A to C in Figure 20(a) is 42.21 ps, which is the same as that from A' to C' in Figure 20(b). This verifies that die-0 is zero skew before die-1 is attached, so the TSV-buffer has done its job. Second, the TG has 14.2 fF capacitance between Node D and the ground, when it is off. This TG completely blocks the clock signal from A to F. When the TG is turned on for the pre-bond testing on die-1, however, it has 108 Ω between its input and output nodes, 16.4 *f*F between its input and the ground, and 18.4 *f*F between its output and the ground. The intrinsic delay of a TG is 1.04 ps. Under this model, the calculated delay from F' to E' is 54.13 ps, which closely matches the simulated delay of 54.14 ps.

3.4.2 Sample Trees

A series of pre-bond testable clock trees are depicted in Figure 21, where the circuit is r_1 from the IBM suite with a TSV bound of 10. The TSVs are shown as black dots, the clock sources as triangles.

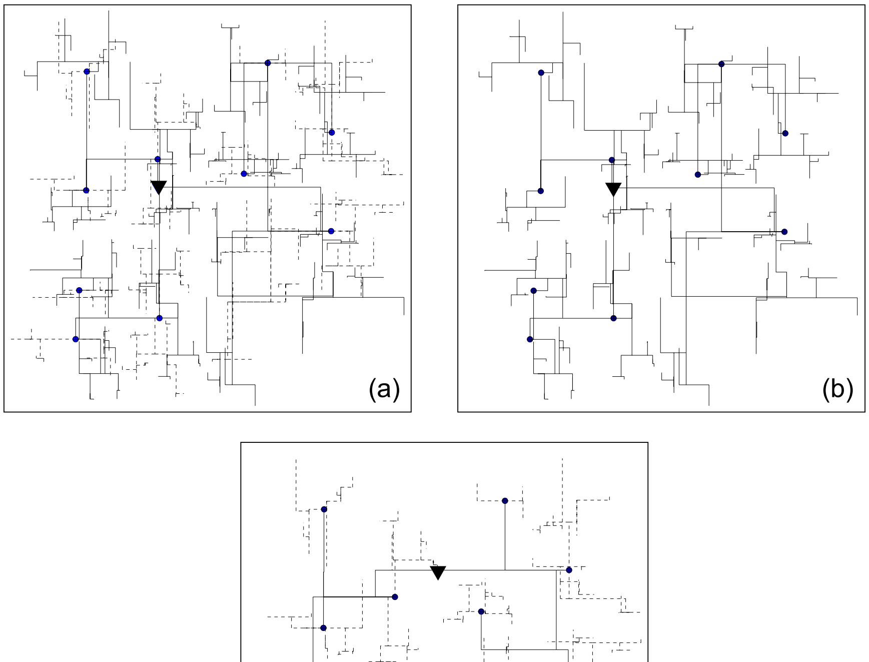


Figure 21: The pre-bond testable clock trees for circuit $r1$ in a two-die stack for a TSV bound of 10. The TSVs and the clock sources are represented by black dots and triangles, respectively. (a) The post-bond 3D clock tree, where the solid and dotted lines denote the trees in die-0 and die-1, respectively. (b) The pre-bond testable 2D clock tree for die-0. (c) The pre-bond testable 2D clock tree for die-1, where the redundant tree and the subtrees are drawn in solid and in dotted lines, respectively.

The zero-skew 3D clock tree for post-bond test and normal operation is shown in Figure 21(a). This 3D clock tree contains 10 TSVs. The solid and dotted lines represent the clock trees in die-0 and die-1, respectively. Note that die-1 contains many subtrees (dotted lines) that are not connected to each other except through die-0. The zero-skew pre-bond testable 2D clock tree for die-0 is shown in Figure 21(b), which is identical to the solid line clock tree in Figure 21(a). The zero-skew pre-bond testable 2D clock tree for die-1 is shown in Figure 21(c), which contains all the subtrees (dotted lines) in die-1 and the redundant tree (solid line) which connects them.

3.4.3 Wirelength, Skew, and Power Results

The wirelength (μm), power consumption (mW), and skew (ps) results are summarized in Table 3, which include the post-bond 3D clock tree (*post-3d*) and the pre-bond testable 2D clock tree for die-0 (*pre-die-0*) and die-1 (*pre-die-1*). For die-1, we report the total wirelength (*WL*) and the wirelength of the subtrees (*WL-sub*), the redundant tree (*WL-red*), and the TG control signal (*WL-TG*). In this case, the wirelength of the pre-bond testable clock tree for die-1 is equal to the sum of *WL-sub* and *WL-red*. In addition, the wirelength of the post-bond 3D clock tree is the sum of the wirelength of *pre-die-0* and *WL-sub* from *pre-die-1*.

Table 3: Wirelength, clock power, and skew results for post-bond testable 3D clock trees and pre-bond testable 2D clock trees.

			post-bond 3D			pre-bond testable								
						die-0			die-1					
ckt	#Sinks	#TSVs	WL	Pwr	Skew	WL	Pwr	Skew	WL	WL-sub	WL-red	WL-TG	Pwr	Skew
r_1	267	57	227141	128.4	13.7	166691	103.0	13.5	150219	60450	89769	62732	68.2	13.0
r_2	598	95	488987	274.1	14.2	328914	196.0	14.1	302023	160073	141950	109031	148.6	11.8
r_3	862	183	616077	361.6	15.5	444156	280.5	15.5	429950	171921	258029	161561	201.9	16.2
r_4	1903	265	1311290	763.2	15.5	889460	536.4	14.9	846980	421830	425151	259442	422.1	15.1
r_5	3101	269	1998950	1115.0	29.1	1255760	715.9	29.1	1236417	743190	493227	310855	615.9	20.9
*ispd*1	121	44	129391	73.3	9.4	99393	64.1	9.2	99169	29998	69171	51214	44.3	6.3
*ispd*2	117	36	127763	71.2	6.8	96093	60.4	6.2	93625	31669	61956	42134	42.0	5.7
*ispd*3	117	42	136676	75.6	5.0	107834	67.0	4.7	101968	28841	73127	52241	45.0	7.3
*ispd*4	91	30	80977	46.8	15.3	61504	40.4	15.2	59870	19473	40397	29449	26.4	14.9
RATIO			1.00	1.00	1.00	0.72	0.79	0.97	0.69	0.28	0.41	0.29	0.57	0.94

Based on the wirelength-related columns, we observe that (1) the total wirelength of *pre-die-0* and *pre-die-1* are comparable (0.72 vs. 0.69 in ratio); (2) in several cases, the wirelength of the redundant tree is about 2x of the total wirelength of the subtrees in die-1 (0.41 vs. 0.28); and (3) in several cases, the wirelength of the TG control signal is about half of the redundant tree in die-1 (0.29 vs. 0.41).

The total clock routing resource cost is equal to the sum of *post-3d* and *WL-red* from *pre-die-1*. Normalizing to the wirelength of *post-3d*, the overall wirelength of the pre-bond testable clock tree and its redundant trees is 1.41. Die-0 and die-1 utilize 51 % and 49 % of the total clock routing resource, respectively. In the post-bond operations, the *post-3d* consumes 71 % of the clock routing resource, which means that 29 % of the clock resource is used for the pre-bond test only. Note that the redundant tree and the TG control signal are used only during the pre-bond testing for die-1. This non-negligible overhead is compensated by the significant power savings to be discussed in Section 3.4.4.

Last, the clock skew values do not exceed 30 ps, satisfying our 3 % of the clock period constraint on the simulated skew. Die-0 consumes more clock power than die-1, primarily because of the TSV-buffers inserted in die-0.

3.4.4 Comparison with The Single-TSV Approach

Our baseline 3D clock tree contains a single, fully-connected zero-skew clock tree in each die; these trees are connected with a single TSV in the two-die stacks and a single column of TSVs in taller stacks. The comparisons of the wirelength (μm) and clock power (mW), and the skew (ps) results from the SPICE simulation are summarized in Table 4. In the multi-TSV designs, we choose the TSV count that gives us the minimum power by an exhaustive search, wherein we sweep the TSV bound from 2 to infinity, construct a 3D clock tree for each bound, and simulate the power consumption. The clock synthesis time for each tree is less than one second in all cases.

We make the following observations. First, our multi-TSV approach significantly outperforms the single-TSV approach in terms of wirelength: 14.8 % to 24.4 % reductions for the two-die stacks, and 39.2 % to 42.0 % reductions for the four-die stacks. Similarly, power savings for the clock trees are 10.1 % to 15.9 % for the two-die cases and 18.2 % to 29.7 % for the four-die cases. These results convincingly demonstrate the benefits of our multi-TSV approach.

Table 4: Comparison between single-TSV and multi-TSV designs.

			Single TSV				Multi-TSV						
												Reduction %	
	ckt	#Sinks	#Bufs	WL	Power	Skew	#TSVs	#Bufs	WL	Power	Skew	WL	Power
Two-die	r_1	267	327	279796	145.0	12.7	57	324	227141	128.4	13.7	18.8	11.4
	r_2	598	693	600880	310.6	12.5	95	684	488987	274.1	14.2	18.6	11.8
	r_3	862	928	765397	404.3	16.1	183	925	616077	361.6	15.5	19.5	10.6
	r_4	1903	1982	1576510	848.7	15.3	265	1963	1311290	763.2	15.5	16.8	10.1
	r_5	3101	2528	2344960	1242.0	22.2	269	2449	1998950	1115.0	29.1	14.8	10.2
	*ispd*09*f*11	121	212	168500	85.4	7.6	44	201	129391	73.3	9.4	23.2	14.1
	*ispd*09*f*12	117	215	164966	84.2	5.8	36	193	127763	71.2	6.8	22.6	15.5
	*ispd*09*f*21	117	226	180867	89.9	9.4	42	211	136676	75.6	5.0	24.4	15.9
	*ispd*09*f*22	91	106	106401	53.2	15.1	30	111	80977	46.8	15.3	23.9	12.1
Four-die	r_1	267	318	272355	141.8	10.5	248	325	160394	111.4	13.3	41.1	21.4
	r_2	598	700	582115	304.5	14.4	434	647	353646	233.9	15.7	39.2	23.2
	r_3	862	945	735299	398.0	14.9	718	922	442903	317.1	13.7	39.8	20.3
	r_4	1903	1956	1532220	831.1	14.8	1651	2011	908375	675.6	16.5	40.7	18.7
	r_5	3101	2939	2312930	1272.0	22.2	2469	3134	1368370	1041.0	20.3	40.8	18.2
	*ispd*09*f*11	121	216	159752	83.1	8.4	129	176	93440	60.0	5.8	41.5	27.8
	*ispd*09*f*12	117	208	155542	80.9	8.9	114	160	90281	56.8	10.2	42.0	29.7
	*ispd*09*f*21	117	212	163816	83.0	17.8	102	160	99179	58.4	7.8	39.5	29.6
	*ispd*09*f*22	91	99	98123	48.7	18.0	81	88	57342	36.1	14.7	41.6	25.9

Second, the total number of buffers (#Bufs) used in the clock trees consists of the clock buffers and the TSV-buffers. Detailed buffer usages in the two-die cases are shown in Table 5, which includes the total number of buffers (#Bufs), the TSV-buffer count (#TBs), and the clock buffer count (#CBs).

Table 5: Buffer usage between the single- and multi-TSV cases. We report the total number of buffers (#Bufs), TSV-buffers (#TBs), and clock buffers (#CBs). The number of dies is two.

	Single TSV			Multi-TSV			
ckt	#Bufs	#TBs	#CBs	#TSVs	#Bufs	#TBs	#CBs
*r*1	327	1	326	57	324	57	267
*r*2	693	1	692	95	684	95	589
*r*3	928	1	927	183	925	183	742
*r*4	1982	1	1981	265	1963	265	1698
*r*5	2528	1	2527	269	2449	269	2180
*ispd*09*f*11	212	1	211	44	201	44	157
*ispd*09*f*12	215	1	214	36	193	36	157
*ispd*09*f*21	226	1	225	42	211	42	169
*ispd*09*f*22	106	1	105	30	111	30	81

We observe that a similar number of buffers is used in both the single- and the multi-TSV trees. In the single-TSV design, buffers are inserted to control the wirelength and slew in each die. In the multi-TSV policy, we need more TSV-buffers to ensure pre-bond testability but use fewer clock buffers. This is because the total wirelength is shorter in the multi-TSV designs and the TSV-buffers have positive impact on slew control.

3.4.5 Impact of TSV Bound on Power

The impact of the TSV bound on wirelength, buffer count, and clock power consumption is depicted in Figure 22. These metrics are normalized to the baseline results from the single-TSV approach. The x-axis corresponds to the TSV bound used to build our multi-TSV pre-bond testable 3D clock trees. Note that the actual TSV usage may be less than the TSV bound because the clock tree synthesis algorithm may determine that the optimal number of TSVs is less than the allowed number. For example, when the TSV bound is set to infinity, only 3097 TSVs are actually used in the four-die stack of benchmark r_5.

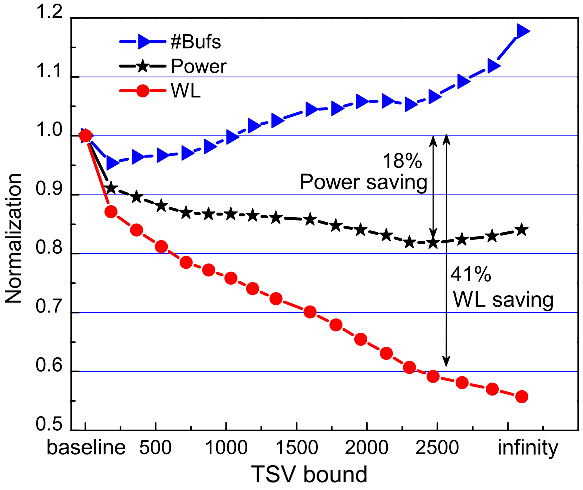


Figure 22: Impact of the TSV bound constraint on wirelength, buffer count, and clock power consumption based on the four-die stack of r_5. The baseline is the single-TSV approach.

We first observe that the wirelength consistently reduces as more and more TSVs are used in our 3D pre-bond testable clock trees. The wirelength savings reach 45 %, if the TSV bound is set to infinity. This confirms that, in general, TSVs help to reduce the overall wirelength of 3D clock trees. Second, the total number of buffers (both clock buffers and TSV-buffers) increases as more TSVs are used, which is mainly due to the insertion of required TSV-buffers for pre-bond testability. Considering both trends, the power consumption decreases consistently but slowly for a time but eventually begins to rise as the cost of the TSV-buffers finally begins to outweigh the wirelength savings. The maximum power saving for r_5 is around 18 %. The corresponding 3D clock tree uses approximately 2500 TSVs across all four dies. With more than 2500 TSVs, the power consumption finally rises because of the excessive number of TSV-buffers. This trend gives us an optimum TSV bound for a given power budget: for the four-die stack r_5, the TSV bound should be set to 300 for a power consumption savings of 10 %.

3.4.6 Impact of CMAX on Power and Slew

The impact of CMAX (the maximum output load each buffer can drive) on skew, maximum rise-slew, and maximum fall-slew among all sinks on all dies is summarized in Table 6. We use four-die stack of benchmark r_1 and compare the single-TSV with our multi-TSV approaches.

Table 6: Impact of CMAX (fF) on skew (ps) and slew (ps) based on four-die stack of r_1. We compare the single-TSV and the multi-TSV approaches.

	Skew		Max rise-slew		Max fall-slew	
CMAX	Single	Multi	Single	Multi	Single	Multi
150	22.6	5.6	37.1	37.4	32.8	33.0
175	22.0	6.3	43.9	44.0	38.7	38.6
200	8.8	6.7	51.5	50.5	45.5	44.3
225	11.3	7.3	58.7	54.0	52.4	47.4
250	9.7	8.3	67.4	59.7	60.1	52.4
275	12.4	11.4	76.4	71.0	68.5	62.5
300	10.5	13.3	86.6	80.8	78.2	71.5

We observe that as the CMAX value increases, the maximum rise and fall slews for both single-TSV and multi-TSV cases increase. In other words, tighter CMAX means better slew. All of the slew values are below the constraint, 10% of the clock period, which is 100ps. The slew values are slightly smaller in multi-TSV designs than single-TSV designs, which is mainly due to (slightly) more buffers inserted for slew control. In terms of skew, the trend is not obvious for the single-TSV case. However, skew tends to reduce with a tighter CMAX value for the multi-TSV case. The main reason is that the wirelength is shorter in these cases, which causes the clock buffers, originally for slew control, to have a positive impact on delay and skew as well.

The impact of CMAX on clock power consumption is plotted in Figure 23. We use four-die stack of r_1 for this experiment. The overall trend is the same in both single-TSV and multi-TSV cases: a tight CMAX results in more power consumption than a loose CMAX. This is because more clock buffers are inserted to meet the tight CMAX constraint. However, the power benefit of the multi-TSV case over the single-TSV case remains consistent regardless of the CMAX value.

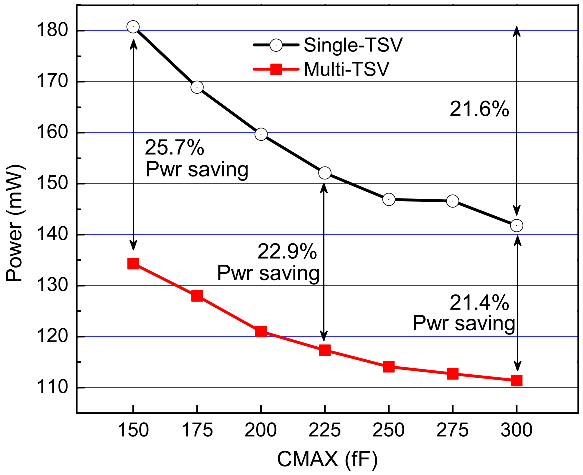


Figure 23: Impact of CMAX (fF) on power consumption (mW) based on four-die stack of r_1.

3.4.7 Trend Study: Impact of TSV Bound and Capacitance

The impact of the TSV capacitance (TSVCap) and the TSV bound on clock power, wirelength, and buffer count (#Bufs) trends is shown in Figure 24. We use the four-die stack implementation of r_5. These metrics are normalized to the results from a design with a single column of TSVs. The TSV capacitance increases from 0 fF to 100 fF. Given both a TSVCap and a TSV bound, we construct a pre-bond testable 3D clock tree, run SPICE simulation on the tree, and report the clock power, wirelength, and buffer count.

We observe that using multiple TSVs affects the clock power in different ways, which depends on the TSV capacitance. First, when the TSV capacitance is small (from 0 fF to 25 fF), we observe that using many TSVs helps to reduce the wirelength, buffer count, and clock power. We obtain the lowest power using 2469 TSVs. In the ideal case when using 0 fF TSVs, we can achieve up to a 23.6 % power reduction compared with the single-TSV case, and wirelength is reduced by more than 42 %. For the 15 fF or 25 fF TSVs, the clock power is reduced by 18.2 % and 14.5 %, respectively.

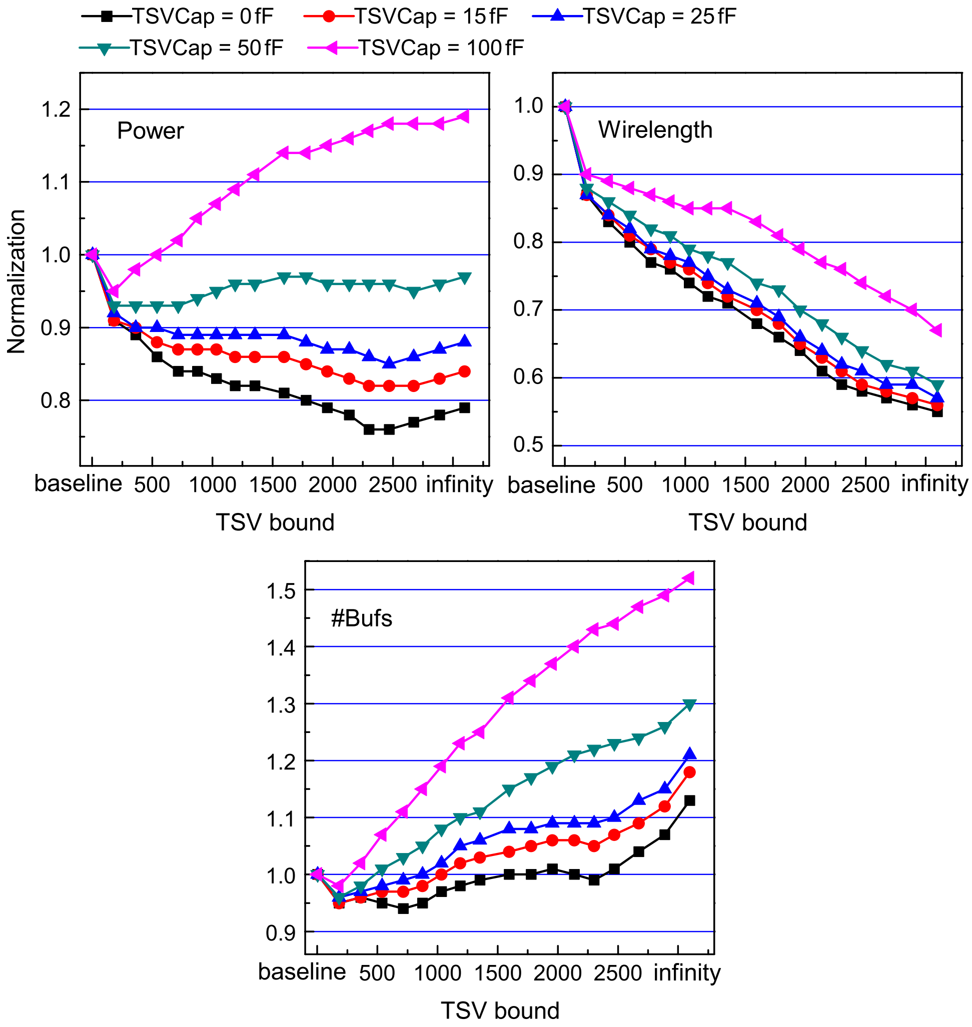


Figure 24: Impact of the TSV capacitance and the TSV usage on the clock power consumption, wirelength, and buffer count trends based on the four-die stack of r_5. The baselines are the single-TSV clock tree for each value of the TSV capacitance.

Second, when the TSV capacitance is large (such as 50 fF or 100 fF), clock power first decreases and then increases when more TSVs are used. In Figure 24, when TSVCap is 100 fF, the lowest clock power (a 4.7 % power reduction) comes from the clock tree with 183 TSVs. When thousands of TSVs are used, the clock power increases significantly.

Third, as the TSV capacitance increases, it becomes more challenging to achieve a low-power clock network. Based on 0 fF TSVs, the multi-TSV policy is able to obtain a low-power design with 23.6 % power saving; for 100 fF TSVs, the multi-TSV strategy can only achieve 4.7 % power reduction.

Those observations are mainly from the following factors. First, the TSV usage and the TSV capacitance have opposite effects on wirelength: using more TSVs tends to reduce the size of each subtree in the non-clock source dies, which reduces the wirelength. However, TSVs with large capacitance tend to unbalance the subtrees, which increases wire snaking. Depending on which factor dominates – the wirelength increase from the large TSV capacitance or the wirelength reduction from multiple TSVs – the trend of the total wirelength changes dramatically. The same discussion applies to the buffer count.

Last, clock power is consumed by the capacitance of the wires, buffers, and TSVs. The multi-TSV strategy helps to reduce the power consumed by the wires but at the cost of increasing the power consumed in the TSVs. When using TSVs with the large capacitance, the TSV power consumption increases faster than wire power consumption decreases, so the total clock power increases. Therefore, as the TSV capacitance grows, the lowest-power design is achieved with just a few TSVs. In general, a large TSV capacitance makes it hard to achieve a low-power pre-bond testable 3D clock tree.

3.5 Summary

In this chapter, we demonstrated how to construct a clock tree for a 3D stacked IC so that both enables test of each die before bonding and provides a minimum-power clock network after bonding. Our solution utilizes many TSVs to reduce wirelength and clock power but necessitates the use of new circuit elements – TSV-buffers and transmission gates – in the clock tree to support the low-skew and low-power characteristics. We studied the impact of buffer insertion on slew rate in 3D stacked ICs clocking. In addition, SPICE results show that our method of inserting multiple TSVs into the clock tree significantly reduces the wirelength and power consumption of the 3D clock tree as compared against a single-TSV baseline. We also studied the impact of the TSV parasitic capacitance on power consumption and wirelength. It shows that a larger TSV capacitance makes it harder to optimize 3D pre-bond testable clock trees.

CHAPTER IV

THROUGH-SILICON-VIA-INDUCED OBSTACLE-AWARE CLOCK TREE SYNTHESIS FOR 3D ICS

TSVs are vertical vias through the silicon die and provide die-to-die communication for multiple functional nets, such as power and ground networks, clock networks, and signal nets. In TSV-based 3D ICs, TSVs create serious blockages for 3D clock routing. As shown in Figure 25.(a), three kinds of TSVs co-exist in 3D designs. Power and ground TSVs (P/G TSVs) usually have large diameter and utilize many local vias to provide the vertical connection in between; signal TSVs and clock TSVs occupy silicon area and have relatively smaller diameter compared with P/G TSVs.

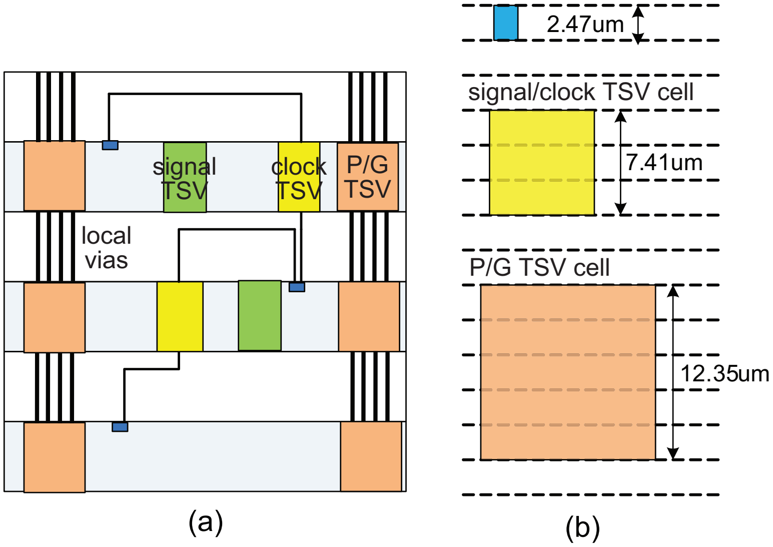


Figure 25: Side and top-down view of via-first power/ground (P/G) TSVs, clock TSVs and signal TSVs. (a) P/G TSVs use many local vias in between vertically, (b) size of the TSV cells (= TSV + keep-out-zone) in terms of the standard cell row height (45nm technology).

Before clock tree synthesis, P/G TSVs and signal TSVs are inserted and occupy both silicon and metal space. The TSV diameters in terms of the standard cell row height in $45nm$ technology is depicted in Figure 25.(b). TSVs are significant layout obstacles due to their large size compared with logic gates and local wires. The TSV-to-gate size ratio is predicted to increase in ITRS 2009, especially when the keep-out-zone around TSVs is taken into account. Therefore, clock routing in 3D IC becomes challenging because these various types of TSVs all become obstacles. Existing work on 3D clock tree synthesis focuses on thermal-aware clock skew minimization [45]. We have also developed 3D clock synthesis for wirelength and power minimization in Chapter 2 and for pre-bond testability in Chapter 3. But, none of these works take into account TSV-induced obstacles.

In this chapter, a practical 3D clock routing problem that stems from TSV-induced obstacles is solved. An analysis on TSV-induced obstacles is performed that the P/G TSVs and signal TSVs are two different types of obstacles in 3D clock routing. A TSV-induced obstacle-aware clock routing algorithm is developed to construct a TSV-overlap-free buffered clock tree. The traditional concept of merging segment is extended to represent clock TSV insertion and clock buffer insertion; two detour policies are presented to handle clock routing in heavily crowded regions. This algorithm is applied on several real benchmarks. The efficiency of the proposed algorithm is demonstrated in the experimental results: the generated TSV-obstacle-aware clock tree does not sacrifice wirelength or clock power too much while avoiding various TSV-induced obstacles.

4.1 TSV Obstacle Analysis

The 3D IC physical design flow consists of several steps. In each design stage, different types of TSVs are added. We use a TSV map to illustrate the size and location of TSVs. Figure 26 shows how the TSV map evolves during each 3D design stage.

During 3D power planning, the 3D power/ground network is constructed, where power and ground TSVs (= P/G TSVs) are inserted at regular locations. To obtain small resistance, P/G TSVs may have a larger size than other TSVs, occupy several standard cell rows, and utilize many local vias to provide the vertical connection in between.

During 3D placement, the locations of gates and signal TSVs are determined. The major reason to insert TSVs during placement is that enough space for the signal TSVs can be reserved, which are several times larger than gates; otherwise, inserting signal TSVs

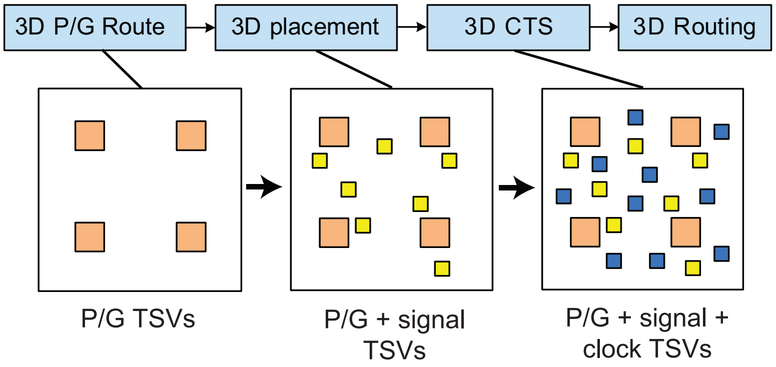


Figure 26: Addition of TSVs during 3D IC physical design. Note that P/G and signal TSVs are added before clock routing.

during routing would create many problems. These P/G and signal TSVs then become obstacles during clock routing[1]. In addition, clock TSVs and buffers are added during clock routing, where clock TSVs themselves become another source of TSV-induced obstacles[2]. The TSV-induced obstacles in 3D clock routing is depicted in Figure 27.

These TSV obstacles behave in the following ways:

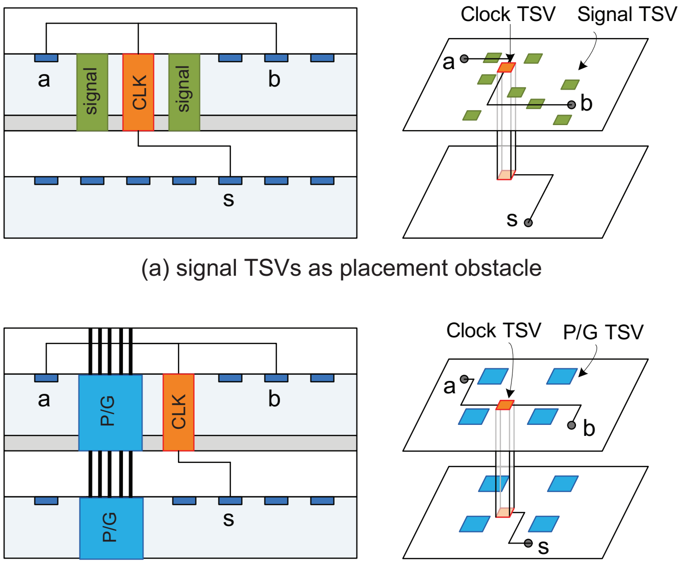
- Signal TSVs: they occupy silicon area only and work as placement obstacles for clock buffers and clock TSVs, which means that, 1) clock TSVs and clock buffers are not allowed to overlap with existing signal TSVs; 2) clock nets are *allowed* to routed over the signal TSVs because their landing pads are in M1 and free up the metal spaces above. An illustration is shown in Figure 27.(a).

- P/G TSVs: they occupy both silicon area and metal layers and thus function as both placement and routing obstacles, which means that, 1) clock TSVs and clock buffers should avoid overlap with existing P/G TSVs; 2) the clock net is not allowed to route over the P/G TSV. An illustration is shown in Figure 27.(b).

- Clock TSVs/buffers: besides P/G TSVs and signal TSVs, 3D clock tree synthesis

[1] In large 3D IC design, 3D global clock synthesis may be performed after floorplanning, where signal TSVs have not been inserted yet. As a result, the TSV-obstacles for 3D clock synthesis include P/G TSVs (acting as both placement and routing obstacles), clock TSVs, and clock buffers. To show the efficiency of our algorithm, this paper focuses on the design flow where 3D clock routing performs after placement, where all types of TSV-induced obstacles exist.

[2] We focus on inserting clock TSVs during clock routing to gain shorter wirelength and lower power. Preserving TSVs before routing is an alternative solution, but out of the scope of this work.



(a) signal TSVs as placement obstacle

(b) P/G TSVs as placement and routing obstacle

Figure 27: TSV-induced obstacles in 3D clock routing for Clock Sinks a, b and s. (a) signal TSVs as placement obstacles, where the clock net is allowed to route over the signal TSVs, (b) P/G TSVs as placement and routing obstacles, where the clock net is not allowed to route over the P/G TSVs.

itself also inserts clock buffers and clock TSVs. They become the same kind of clock routing obstacles as signal TSVs if added in an iterative fashion such as DME-based clock tree embedding.

Due to the sheer size of TSVs, detour policies are required to handle the cases when TSV obstacles significantly block the routing and placement area. A sample of the buffered 3D clock tree is plotted in Figure 28, which avoids overlap with TSV obstacles.

4.2 Preliminaries

4.2.1 Problem Formulation

The formal definition of *TSV-induced obstacle-aware 3D clock routing problem* is as follows: Given a 3D TSV obstacle map consisting of signal TSVs and P/G TSVs on each die, a set of clock sinks on each die, dimensions of clock buffers and clock TSVs, an upper bound on TSV count, the objective is to construct an overlap-free buffered 3D clock tree such that 1) clock skew is zero; 2) clock wirelength and power are minimized; 3) clock slew is bound under the given constraint. *The overlap-free constraint* requires that 1) clock buffers and

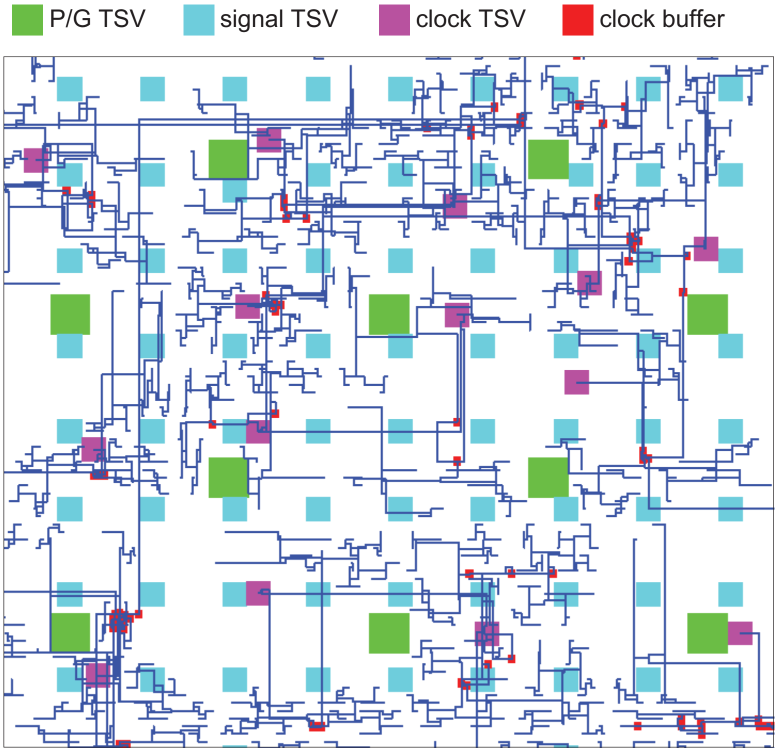


Figure 28: TSV-obstacle avoidance in 3D clock routing. TSVs cannot overlap with each other, clock buffers cannot overlap with TSVs, and clock nets cannot route over P/G TSVs.

clock TSVs do not overlap with the signal TSVs and P/G TSVs; 2) clock nets are not routed over the P/G TSVs.

4.2.2 Extension of Merging Segment Concept

In our 3D TSV-obstacle-aware clock routing, we extend the concept of merging segment (ms) that is primarily used for clock internal nodes only to denote the candidate locations of non-zero-sized clock buffers and clock TSVs under minimum skew and wirelength objectives. Specifically, $msp(p)$ and $msc(t)$ denote the ms of an internal Clock Node p and the center of a non-zero-sized clock TSV or buffer, respectively. The extended merging segment concept is illustrated in Figure 29.

We focus on via-first TSV-induced obstacles in 3D clock tree synthesis[3]. These obstacles can be classified into two types: *placement obstacle* that blocks the silicon area and affects clock buffer/clock TSV insertion. This obstacles comes from P/G TSVs, signal TSVs, and

[3] We apply our TSV-obstacle-aware clock routing on via-first TSV 3D application. It can be easily extend to via-middle or via-last TSVs.

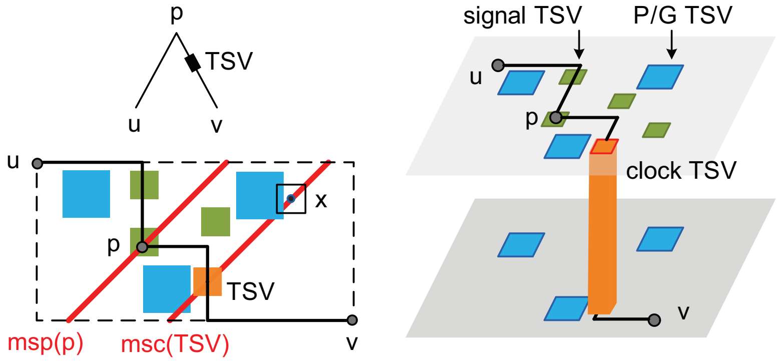


Figure 29: Illustration of the extended merging segment concept. When merging Nodes u and v in different dies, $msp(p)$ denotes the merging segment of Node p; $msc(TSV)$ denotes the center-point locations of the clock TSV. Signal TSVs allow Node p and the clock net to route over it. However, clock TSV x cannot overlap with a P/G TSV.

clock TSVs; *Routing obstacle* that blocks the routing area and affects the clock routing topology. This obstacle comes from P/G TSVs.

We use the following merging segments in this work:

- *Placement-overlap-free merging segments*: collection of the merging points that the corresponding non-zero-sized clock components, *i.e.*, clock TSVs and clock buffers, have no overlap with the placement obstacles and have its center point located along the *msc*.

- *Routing-overlap-free merging segments*: potential location of *msc* and *msp*, which are able to reach the children merging segments with the minimum distance while avoiding routing obstacles.

- *Feasible merging segments*: for *msp*, the feasible merging segment becomes the routing-overlap-free merging segments; for *msc*, the feasible merging segment satisfies both placement-overlap-free and routing-overlap-free requirements.

4.3 Overview of the algorithm

Our TSV-obstacle-aware clock routing algorithm consists of the following two steps:

Bottom-up feasible merging segment construction: Our goal is to determine the feasible merging segments (= FMS) for internal nodes, clock buffers, and clock TSVs.

Depending on the merging types (*e.g.,* wire merging, TSV merging), the flow is different. When merging $ms(u)$ and $ms(v)$ to $msp(p)$, we first generate the $msp(p)$ under a zero-skew constraint and determine its FMS using the *Nine-Region-Based Cutting* method explained in Section 4.4.2. Note that clock TSVs and clock buffers may be inserted along Edges (u, p) or (v, p) together. If a clock TSV is required on (v, p) because u and v are in different dies, we aim to find the FMS for p and the TSV. When a clock buffer is required to be inserted along (u, p) or (v, p), our goal is to determine the FMS for p using the *Nine-Region-Based Cutting* method and for the buffer and TSV using both the *Nine-Region-Based Cutting* and *Expanded-Obstacle Cutting* method explained in Section 4.4.2 and Section 4.4.1. If no FMS can be found in the merging area with the shortest distance, we utilize two detour policies for both placement obstacles and routing obstacles explained in Section 4.5.

Top-down obstacle-aware embedding: Our goal is to decide the exact embedding point along the FMS and to determine the clock routing topology. The embedding points for the clock buffers and clock TSVs should avoid overlap between other clock TSVs, P/G TSVs, signal TSVs, and clock buffers. In addition, we use the *Nine-Region-Based* method to determine the final routing-overlap-free topology.

4.4 Feasible Merging Segments

We present two techniques to obtain overlap-free merging segments: *Expanded-Obstacle Cutting* to obtain a placement-overlap-free merging segment and *Nine-Region-Based Cutting* to determine a routing-overlap-free merging segment. Based on whether a given TSV is a placement or routing obstacle, we apply different cutting policies. For P/G TSVs, both the *Expanded-Obstacle Cutting* and *Nine-Region-Based Cutting* methods are used; for signal TSVs, only the *Expanded-Obstacle Cutting* method is used.

4.4.1 Expanded-Obstacle Cutting

The goal of *Expanded-Obstacle Cutting* method is to determine the placement-overlap-free merging segment of clock TSVs and buffers. The outcome is an insertion of a clock TSV or a clock buffer with the center point located along the merging segment. In this case, this clock TSV or the clock buffer should have no overlap with other placement obstacles such as P/G TSVs, signal TSVs, other clock TSVs, and clock buffers that are already existing in the layout.

Given an initial merging segment $msc(t)$, the dimension of the clock component, d, to be added (= either clock TSV or clock buffer), and a placement obstacle $obst$, the basic procedure of *Expanded-Obstacle Cutting* is as follows: We first construct an expanded-overlap-free boundary (EOFB) by expanding the $obst$ by the distance of $d/2$ in all four directions. We then utilize EOFB to determine a feasible merging segment: any merging point along the $msc(t)$ outside the EOFB is a placement-overlap-free point; in other words, any point along $msc(t)$ inside the EOFB will have overlap with the placement obstacle. Figure 30 shows an illustration.

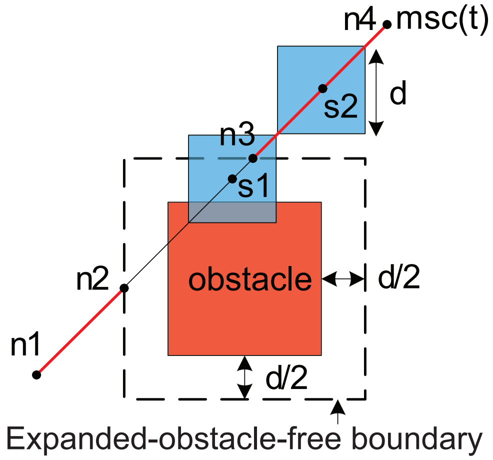


Figure 30: Expanded-Obstacle Cutting on a merging segment $msc(t)$. The expanded-overlap-free boundary determines that Segments $n1$-$n2$ and $n3$-$n4$ are the feasible merging segments. A clock TSV with $s1$ as the center will cause an overlap with the obstacle, whereas inserting the TSV with its center on $s2$ is safe.

4.4.2 Nine-Region-Based Cutting

Given a merging segment of Child u, our goal in *Nine-Region-Based Cutting* is to find the routing-overlap-free feasible merging segments of its Parent p (= either an internal clock tree node, clock TSV, or clock buffer), so that the merging segment of p provides a feasible routing topology to its child u with the shortest distance.

A routing obstacle (in red) partitions the routing area into nine regions with its four extended boundary lines as shown in Figure 31.(a). These nine regions are used to determine the connectivity between a pair of nodes. For instance, Node p can connect to u in both HV (horizontal first, then vertical) and VH (vertical first, then horizaontal) topology, whereas VH routing type from Nodes p' to u' is blocked by the obstacle.

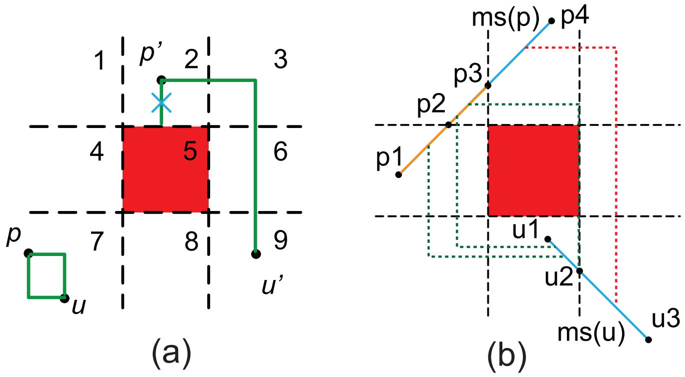


Figure 31: Nine-Region-Based Cutting method. (a) Nine regions partitioned by a routing obstacle in red. p to u is HV and VH connectable, and p' to u' is HV only. (b) $(p1, p2)$ and $(p2, p3)$ are the routing-overlap-free merging segment of $ms(p)$ to its child $ms(u)$, $(p3, p4)$ is not due to the shortest distance constraint.

Note that these nine regions are symmetric; they can be classified into three groups: Group A: Regions 1, 3, 7, and 9; Group B: Regions 2, 4, 6, and 8; and Group C: Region 5. The connectivity of merging segments can be easily determined by referring to the region groups. (1) When p is located in Region 2, it is two-way (both HV and VH) connectable to Regions 1, 2 and 3; HV connectable to Regions 4, 6, 7 and 9; and is not connectable to Regions 5 and 8. The same discussion applies to the case when p is located in Regions 4, 6, or 8. (2) When p is located in Region 1, it is two-way (both HV and VH) connectable to Regions 1, 2, 3, 4, 7 and 9; HV connectable to Region 6; VH connectable to Region 8; and is not connectable to Region 5. The same discussion applies to the case when p is located in Regions 3, 7, or 9. (3) When p is located in Region 5, it is not connectable to any region.

Given the merging segments of Parent p and its Child u, the *Nine-Region-Based Cutting* method consists of two steps. First, it constructs nine regions for a routing obstacle. Correspondingly, a merging segment is divided into several sub-segments, where each sub-segment belongs to a unique region. Second, it checks each sub-segment of Parent p to see if it is connectable to any sub-segment of Child u. In addition, the distance between these two sub-segments should be equal to the shortest distance between $ms(p)$ and $ms(u)$. If these conditions are satisfied, the current sub-segment of the parent is routing-overlap free. Figure 31.(b) shows a sample, where $(p1, p2)$ and $(p2, p3)$ are the FMS of $ms(p)$. Note that $(p3, p4)$ is not a FMS since the distance between $(p3, p4)$ and $(u2, u3)$ is longer than the shortest distance between $ms(p)$ and $ms(u)$. This technique also helps us determine the actual routing topology during the top-down embedding procedure when a routing obstacle presents in the merging region.

4.5 TSV-Obstacle-Aware Detouring

In this section, we discuss two major cases when no feasible merging segment exists within the merging region of the shortest distance: one is for routing obstacles, the other is for placement obstacles. We develop two detour policies to find the feasible merging segments outside the merging region.

4.5.1 Routing-Obstacle-Aware Detour

When a routing obstacle blocks the routing region, we use the routing-obstacle-aware detour technique to find the feasible merging segments outside the merging region. This situation is usually caused by big P/G TSVs. In this case, the merging segment of the parent is not connectable to that of its children $ms(u)$ and $ms(v)$. Two detour cases are shown in Figure 32, when the merging segment is a point (Figure 32.(a)) or an arc (Figure 32.(b)).

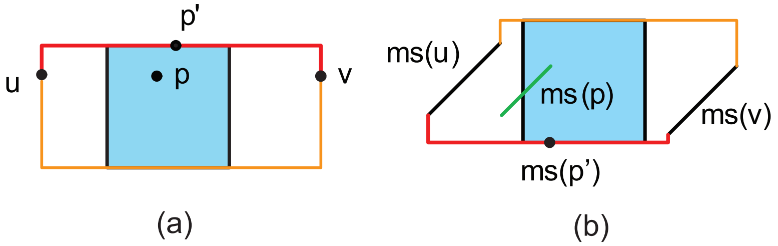


Figure 32: Detour policy when a routing-obstacle blocks the routing region. (a) merging segment for u and v are points, where the top (= red) detour is chosen over the bottom (= orange), (b) merging segments for u and v are lines, where the bottom (= red) detour is chosen.

A new merging point location of Parent p' is chosen along the boundary of the obstacle that the u-to-p'-to-v wirelength (L') is minimized. We then calculate the merging distance between Nodes p', u and p', v based on the zero-skew equations. In this case, L' becomes the shortest distance to connect $ms(u)$ and $ms(v)$ while traveling along the obstacle boundary. And p' is the zero-skew point in between u and v.

4.5.2 Placement-Obstacle-Aware Detour

We use a TSV merging example to show how our detour policy works in the case that no feasible merging segment exists when inserting a clock TSV. When merging Node a in the top die and b in the bottom die at their merging segment located on the top, the expanded-overlap-free boundary of the signal TSV (= placement obstacle) may cover both Nodes a and b. As a result, the placement-overlap-free merging segment of the clock TSV to be added exists in the merging region. The placement-aware detour for TSV merging is shown in Figure 33.

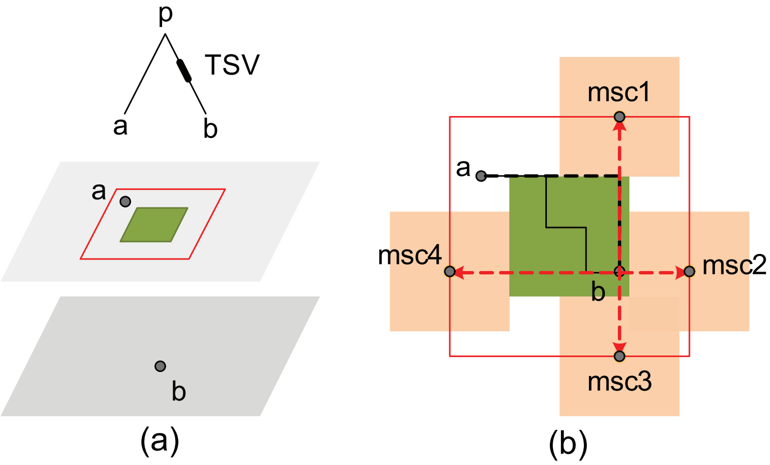


Figure 33: Placement-obstacle-aware detour for TSV merging. A signal TSV occupies the merging area between Nodes a and b where a TSV is needed. A feasible merging segment for this clock TSV is added on the expanded-overlap-free boundary with the shortest merging distance. $msc1$-$msc4$ show four candidates. We choose $msc2$ due to its shortest distance to b.

As shown in Figure 33.(a), b in the bottom die is allowed to have overlap with the signal TSVs in the top die. Referring to the top-down view in Figure 33.(b), our detour policy is to extend Node b in four directions, and obtain intersections along the expanded-overlap-free boundary of the obstacle, *i.e.*, $msc1$ to $msc4$, which are the four potential locations of the merging segments of the clock TSV. We choose $msc2$, which is the nearest intersection to b, update the distance between $msc2$ and b, and the merging distance between a and $msc2$.

By solving the zero-skew equations, we can then determine the merging segment of Parent p.

4.6 Clock TSV Merging

We observe that a longer feasible merging segment helps avoid TSV-induced overlap with clock buffers and other TSVs. Our policy is to find the longest feasible merging segment by sweeping the distance between the clock TSV and its child. An illustration on clock TSV merging is shown in Figure 34.

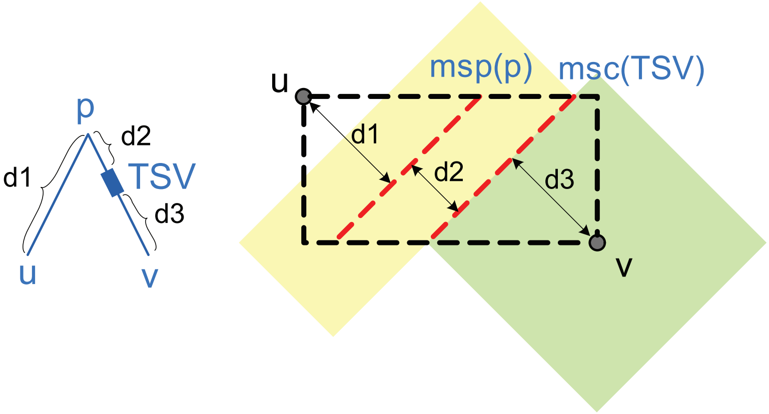


Figure 34: Finding the longest feasible merging segment for the clock TSV by sweeping the distance between clock TSV and $ms(v)$.

We first determine the merging segment for the clock TSV, denoted $msc(TSV)$, using $d3$, which is the distance between $msc(TSV)$ and $ms(v)$, the merging segment of Child v. We then apply the *Nine-Region-Based Cutting* and *Expanded-Obstacle Cutting* methods to decide the feasible merging segment for the clock TSV. After that, we determine the merging segment of Parent p, denoted $msp(p)$, by deriving Distances $d1$ and $d2$ based on the conventional zero-skew constraint and the shortest merging distance requirement. The feasible merging segment for p is determined using the *Nine-Region-Based* Cutting method in this case.

To find the longest $msc(TSV)$, we sweep Distance $d3$ with a certain step (such as $d/2$). Under a given $d3$, we obtain a pair of the feasible merging segments for the TSV and p. We choose the longest length of the feasible merging segment for the TSV as our final merging solution. This scheme is shown to provide a better chance to avoid overlap between clock

Table 7: Benchmark information. Footprint area is in μm^2.

Ckt	Area	#Sinks (die-0 + die-1)	#Signal TSVs	#P/G TSVs
IDCT	433^2	117 + 356	342	82
8086	420^2	230 + 427	323	82
8051	400^2	347 + 1009	306	64
b18	483^2	1652 + 1448	440	100
b19	590^2	3099 + 3071	462	144

buffers and clock TSVs during the top-down embedding.

4.7 Experimental results

4.7.1 Simulation Setting

We apply our TSV-obstacle-aware clock routing method to the IWLS 2005 benchmarks [88], as listed in Table 7. We use 45nm technology. The P/G TSV cell is 12.35μm × 12.35μm, the signal and clock TSV cells are 7.41μm × 7.41μm. And the clock buffer cell occupies 2.09μm × 2.47μm.

For each benchmark, we perform 3D power planning and gate/TSV placement using a Cadence Encounter-based 3D physical design tool-chain. We obtain a TSV obstacle map (including P/G and signal TSVs) and clock sinks locations. The benchmark information is summarized in Table 7. We then apply our TSV-obstacle-aware 3D clock routing algorithm to achieve an overlap-free 3D clock tree under the given maximum slew rate, TSV count bound, and zero-skew (under the Elmore delay model) constraints. Then we apply SPICE simulation on entire 3D clock network to report clock power consumption and timing. The clock frequency is set to 1GHz, with the supply voltage 1.1V. The maximum clock skew from the simulation is required to be under 30ps. The maximum loading capacitance for each clock buffer is 100fF. TSV capacitance is 15fF, and resistance is 35mΩ. The clock source is located on the topmost die (= die-0).

4.7.2 Sample TSV-Aware Clock Topology

The 3D clock routing result of benchmark $b19$ is shown in Figures 35 and 36, where the first ignores TSV obstacles (= Figure 35) and the second avoids TSV obstacles (= Figure 36).

Both results are based on the same set of two-die stack clock sinks, TSV obstacle map, and clock constraints. We highlight several dense regions in the tree and show the details

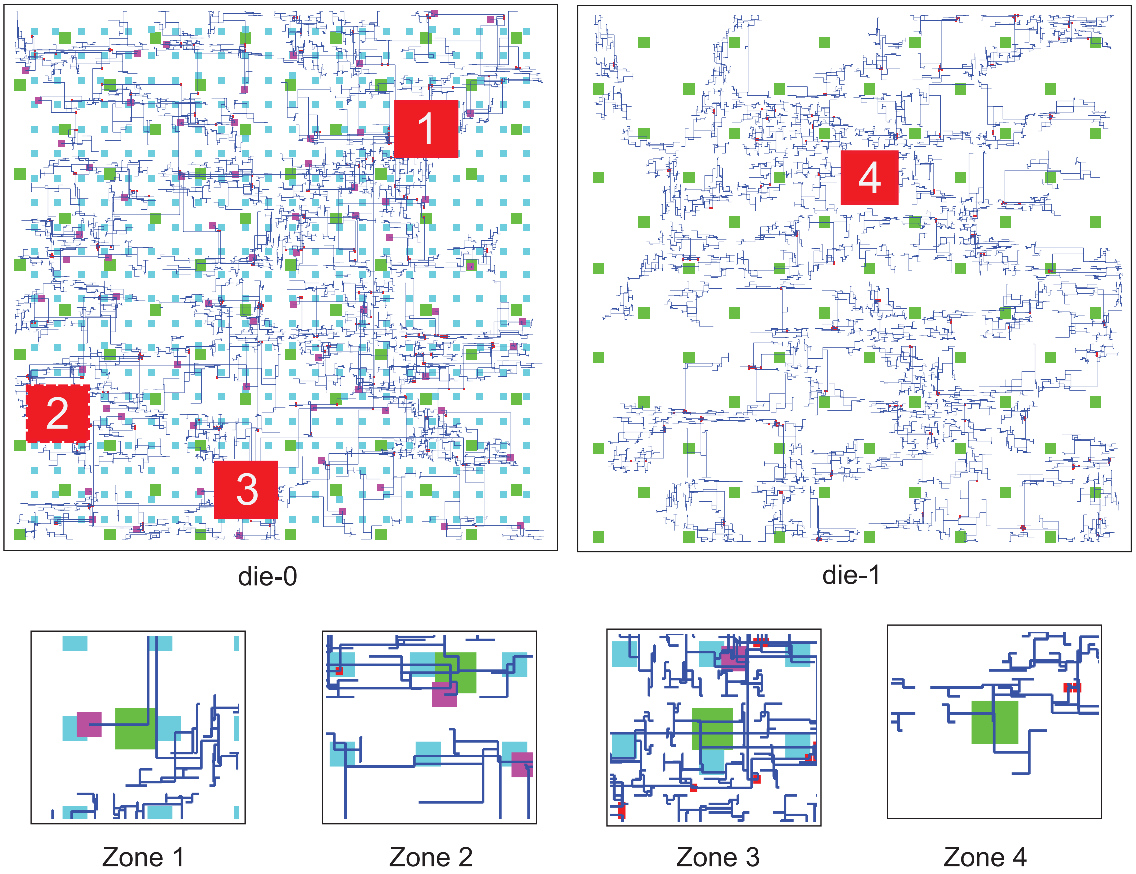


Figure 35: A two-die stack clock routing WITHOUT considering TSV obstacles. We show P/G TSVs (green), signal TSVs (blue), clock TSVs (red), clock wires, and clock buffers (red). This tree violates several overlapping constraints, including clock TSVs overlap with other P/G TSVs, signal TSVs, and buffers, and routing over P/G TSVs.

inside. We observe that many violations (= illegal overlaps) occur, especially in the dense regions, including clock TSV overlap with other P/G and signal TSVs, routing over P/G TSVs, buffer and signal TSV overlap. However, by using our TSV-obstacle-aware clock routing algorithm, we see that no clock net is routed over the P/G TSVs, and no overlap exists among P/G TSVs, signal TSVs, clock TSVs, and clock buffers.

4.7.3 Impact of TSV-Induced Obstacles

We compare the quality of the clock trees with and without TSV obstacle avoidance to quantify various kinds of overhead that occur by avoiding TSV obstacles. A comparison is shown in Table 8, which includes wirelength, clock skew, clock slew and clock power under the same amount of clock TSVs and the same clock buffer used. We also show the

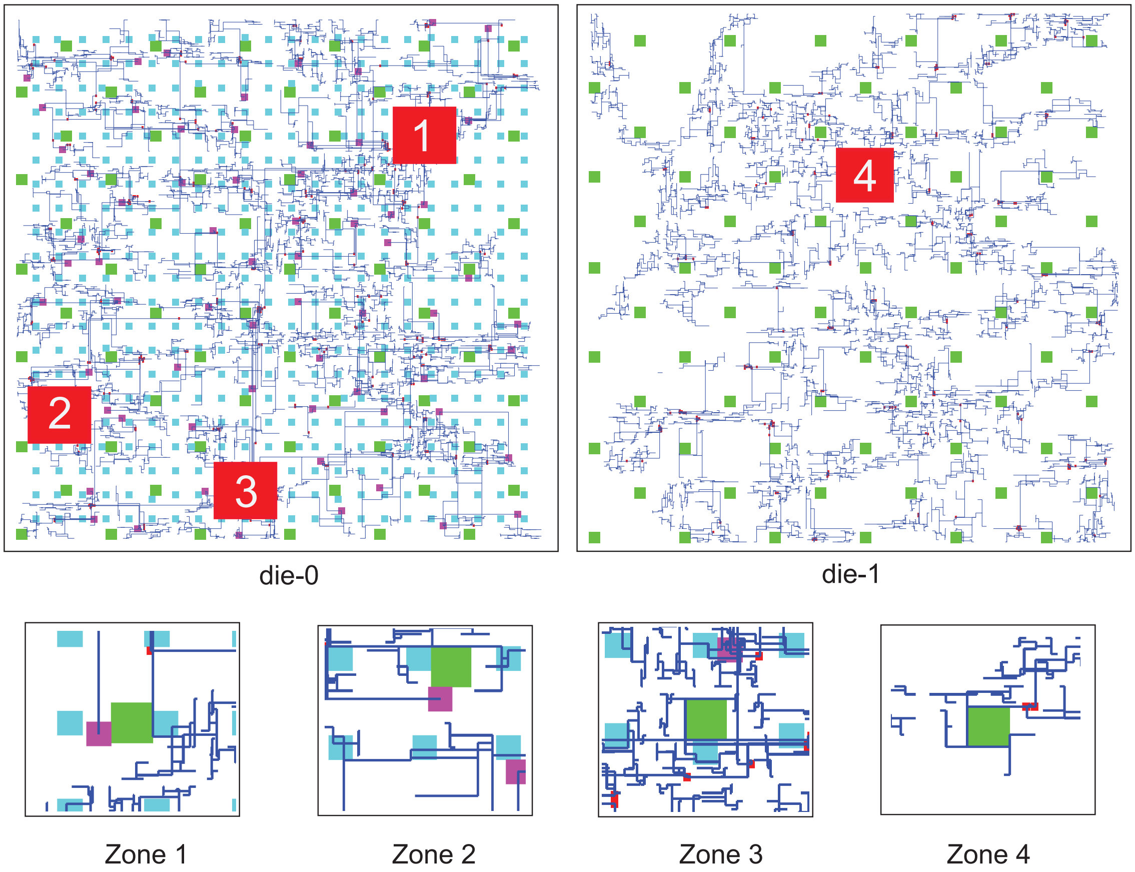


Figure 36: A two-die stack clock tree WITH TSV obstacle avoidance for the same circuit as Figure 35. This tree does not contain any illegal overlap.

percentage increase of wirelength and power of the TSV-obstacle-aware clock routing results over the obstacle-ignoring cases.

First, our TSV obstacle-aware clock routing algorithm is able to achieve a TSV-overlap-free clock tree. Second, the clock skews are all zero under the Elmore delay model and are well controlled under 30ps from SPICE simulation. Third, our TSV-obstacle-aware clock routing results are comparable to the result when TSV obstacles are ignored. We show two cases of clock TSV usage, one that uses a small number of TSVs, the other one that uses a larger number of TSVs. In most of the cases, the TSV-obstacle-aware clock tree has slightly larger wirelength or clock power; in some benchmarks, TSV-obstacle-aware clock routing obtains slightly better results. This phenomenon demonstrates that our TSV obstacle avoidance method works well while keeping the overhead almost negligible. Moreover, the runtime of our TSV-obstacle-aware clock routing is within several seconds.

Table 8: Comparison of two 3D clock routing results. The first one avoids TSV obstacles by applying TSV-obstacle-aware routing; and the second one ignores TSV obstacles. We also show % increase of clock power and wirelength of TSV-obstacle-aware routing.

		Avoid TSV obstacles				Ignore TSV obstacles				Increase (%)	
ckt	#TSVs	WL (μm)	#Bufs	Pwr (mW)	Skew (ps)	WL (μm)	#Bufs	Pwr (mW)	Skew (ps)	WL (μm)	Pwr (mW)
Using small amount of clock TSVs											
IDCT	4	21431	178	16.6	17.7	21810	178	16.7	19.7	-1.7	-0.3
8086	4	25055	131	12.3	19.3	25223	125	12.0	22.4	-0.7	3.1
8051	5	50128	456	41.5	15.0	47616	449	40.8	15.4	5.3	1.6
b18	10	75653	471	42.4	23.9	74964	468	42.1	20.4	0.9	0.9
b19	10	160621	823	81.8	16.4	158082	818	81.0	15.7	1.6	1.0
Using more clock TSVs											
IDCT	10	22069	172	16.7	19.7	22146	170	16.5	18.5	-0.3	0.9
8086	14	22475	110	10.7	21.1	22459	110	10.7	21.1	0.1	0.0
8051	35	51627	439	42.2	11.7	52007	438	42.2	11.6	-0.7	-0.1
b18	58	91273	517	50.7	14.2	90811	515	50.5	14.3	0.5	0.4
b19	70	146603	792	78.2	20.9	145798	791	78.0	20.7	0.6	0.2

4.8 Summary

In this chapter, we addressed a practical obstacle issue in TSV-based 3D clock tree synthesis and studied how to avoid TSV-induced obstacles in 3D clock routing. We first discussed how power/ground TSVs (P/G TSVs) and signal TSVs become two different types of obstacles in 3D clock routing. We then developed a TSV-obstacle-aware clock routing algorithm to construct a TSV-overlap-free buffered clock tree. We proposed a TSV-obstacle-aware DME technique. We also studied how to apply detour when no feasible merging segment exists. Experiments show that we can achieve a buffered clock tree that avoids overlapping with TSV-induced obstacles while keeping the wirelength and power overhead to a minimum.

CHAPTER V

TSV ARRAY UTILIZATION IN LOW-POWER 3D CLOCK NETWORK DESIGN

5.1 Introduction

Three-dimensional ICs (3D ICs) is one of the most promising technologies that enables higher integration and further miniaturization. However, through-silicon vias (TSVs), may cause reliability and cost issues that delay mainstream acceptance [4, 58]. TSVs can squeeze or stretch adjacent transistors and interconnects. This material deformation may lead to mobility change and thus performance variation [4, 59]. It also causes mechanical reliability issues, causing open hole, short, or even crack. TSV-to-TSV and TSV-to-device coupling affect timing and signal integrity [60–62]. All these TSV-related issues require extra design efforts.

TSV array, defined as a group of TSVs placed in regular positions either in 1D or 2D grid fashion, is shown to be more manufacturable and practical to address the TSV-related reliability issues. As shown in Figure 37, multiple TSV arrays can be found in block-level design. It is also possible that a single 2D TSV array covers the entire layout area in gate-level design. Recent studies show that placing TSVs at any desired locations during placement [63] or routing [64] leads to shorter wirelength and better timing results compared with regular locations (TSV arrays). However, this irregular placement may result in TSVs crowded in a certain region and cause problems in coupling [61,65], timing variations [62,66], and mechanical reliability [59, 67].

In previous chapters, we developed 3D clock synthesis methods for clock power minimization in Chapter 2 and for pre-bond testability in Chapter 3. Existing works on 3D clock synthesis also focus on variability analysis [1] and fault tolerance [89]. However, none of them consider TSV arrays, but all insert TSVs at any desired positions during routing.

Simply extending existing work for TSV array design style cannot guarantee power efficiency. In the TSV array 3D design, the TSV count and locations have been determined BEFORE clock routing, where any TSV movement or additional TSV insertion is not allowed. Consequently, the clock network is limited to utilize these given TSVs. The final clock power is significantly affected by the TSV array utilization (how many and where). Thus, a practical question is *what is the optimal TSV array utilization for power efficient 3D clock network design with skew minimization and slew constrained?*

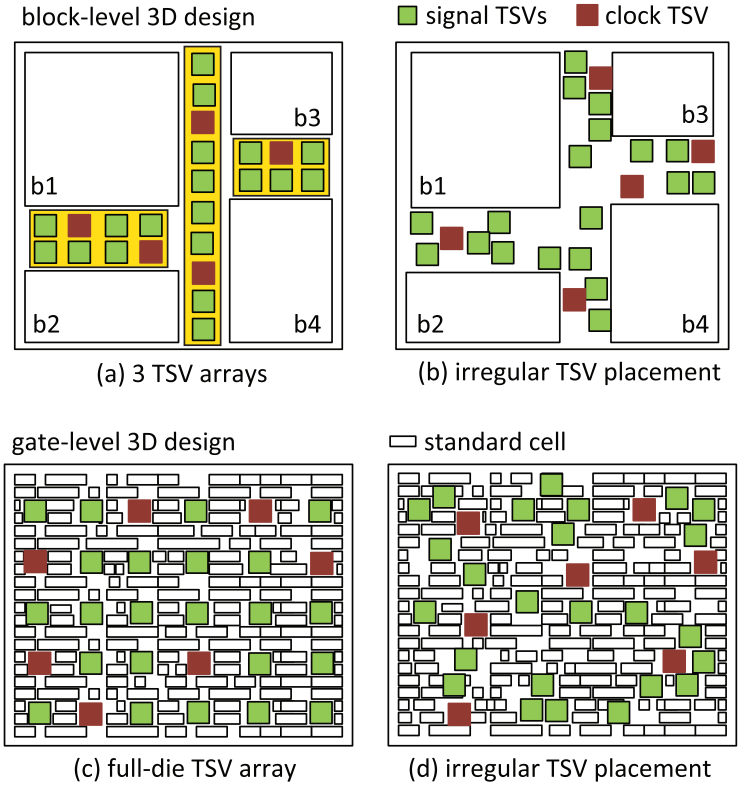


Figure 37: TSVs at regular locations (TSV arrays) vs. irregular locations in block-level and gate-level 3D designs.

This chapter addresses the 3D clock routing problem for TSV array utilization to construct low-power and reliable 3D clock networks. A novel method named decision-tree-based clock synthesis (DTCS) is presented to generate small-skew and low-power clock trees by efficiently exploring the entire solution space for the best TSV array utilization. The existing 3D clock synthesis method is also extended for TSV array utilization. At last, The efficiency of the proposed DTCS method is verified for both gate-level chip-scale 3D clock designs and block-level global clock designs. Close-to-optimal solutions of power efficiency can be figured out in short runtime.

5.2 Clock Design Methodology for TSV Arrays

5.2.1 Problem Formulation

Given a set of clock sinks and an $M \times N (= K)$ TSV array for each die in a 3D stack, the goal of 3D Clock Routing with TSV Arrays (3D-CRTA) problem is to build a buffered 3D clock tree while using up to K TSVs for each die so that clock skew and power are minimized under clock slew constraint. If an additional constraint that the TSV bound $B < K$ is given on the total TSV count, the clock tree should not use more than B TSVs in each die.[1] In both cases, we do not add additional TSVs into the arrays.

5.2.2 Overview

We develop a **D**ecision-**T**ree-based **C**lock **S**ynthesis (**DTCS**) method to construct a low-power 3D clock tree with minimum skew while satisfying the slew and TSV count constraints. This method explores the entire solution space of TSV array utilization to find out a power-optimal design based on a decision tree. Meanwhile, the clock routing and buffering method [1] is integrated into DTCS method so that the clock topology is balanced and buffer insertion is performed to satisfy skew and slew constraints.

Our DTCS method consists of three steps. (1) *Decision Tree Construction*: We generate a decision tree that contains all the feasible solutions of TSV array utilization. The decision tree evaluates clock power overhead in various 3D clock trees under consideration. (2) *Clock Tree Construction*: We construct an initial 3D clock tree with lowest power under no TSV bound constraint. If no TSV count constraint is given, this initial tree becomes the final tree, and our algorithm terminates at this point. (3) *Clock Tree Refinement*: If the initial clock tree exceeds a TSV bound, we remove some TSVs by modifying the cut orientation of some decision nodes. Meanwhile, the resulting power and skew are kept minimal.

To verify the efficiency, our DTCS method focuses on finding out the optimal TSV

[1] A practical purpose of this TSV count constraint B is to reserve $K - B$ TSVs in each array for signal and/or P/G routing later.

utilization, but tries to keep using existing routing and buffer insertion techniques [1]. We use two-die stack to describe the proposed method and algorithms, which is then extended to handle more-than-two dies in Section 5.3.4.

5.2.3 Our Decision Tree

Our decision tree, as shown in Figure 38, is represented as a binary tree which visualizes the entire solution space for TSV array utilization in low-power 3D clock design. A decision node d_i (shown in a gray box) for a sink set S_i contains the following information: (1) cut orientation of S_i (z_i is 0 for XY-cut, 1 for Z-cut); (2) a sink set S_i^{XY} after an XY-cut is applied onto S_i, where $S_i^{XY} = \{S_{2i}, S_{2i+1}\}$ that contains two children nodes; (3) a sink set S_i^Z after a Z-cut is applied onto S_i; (4) power and TSV utilization for S_i. The root node d_1 contains the original 3D sink set, and the leaf nodes represent the subsets that have a unique solution.

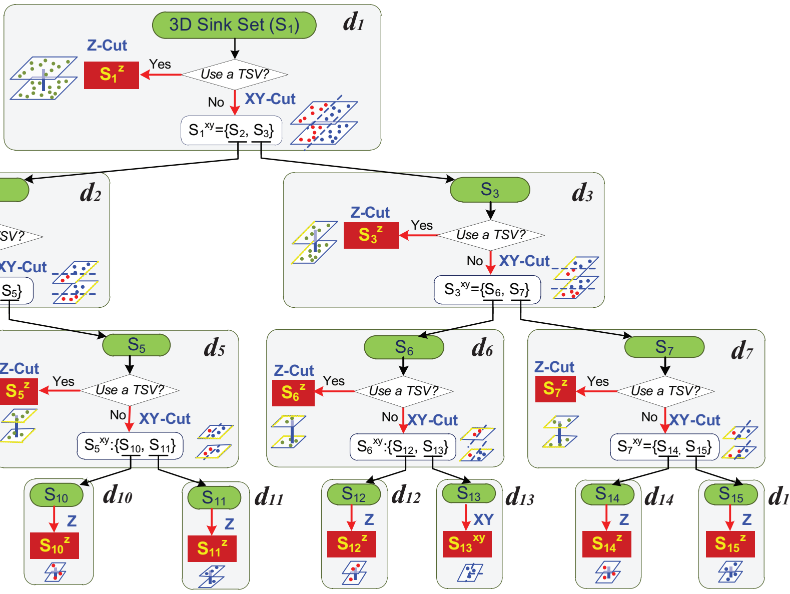


Figure 38: Illustration of our decision tree that shows the entire solution space of TSV array usage for low power. Each node (except leaf nodes) can choose between using one TSV (= Z-cut) or multiple TSVs (= XY-cut) in the array. Once the entire decision tree is built, we obtain different 3D clock trees by visiting all possible sink-to-root paths during our clock tree construction step.

A decision tree is built in a top-down recursive fashion, where we explore all feasible partitioning options, both Z-cut and XY-cut, for a given subset of clock sinks. XY-cut and Z-cut are two partitioning steps to define the clock abstract topology. An *XY-cut* partitions the sinks based on their X or Y coordinates, where the 3D sinks are flattened into 2D. The median value of the given sink set is the cutline for the XY-cut. An XY-cut results in multiple TSVs to connect descendant sink sets. A *Z-cut* separates the sinks die-wise so that the sinks in the same die are assigned to the same subset. A Z-cut immediately requires one TSV for connecting two subsets. This top-down partitioning and decision tree construction continue until the current sink set has a unique TSV utilization.

In our decision analysis, we call a sink set *unique* if it does not require any further partitioning and exploration of TSV utilization. We define a unique set based on the following conditions (shown in red rectangles in Figure 38). *Condition 1*: S_i is a 2D sink set (e.g., S_9 in Figure 38); *Condition 2*: S_i is a 3D sink set, but it requires a Z-cut due to the limited availability of TSVs in the TSV array; *Condition 3*: S_i^Z is obtained by applying a Z-cut on the 3D sink set S_i that does not satisfy Condition 2 (e.g., S_1^Z-S_7^Z in Figure 38). The second condition requires us to look ahead one more partitioning level down: when the bounding box of S_i contains only one TSV in the array or at least one of the two subsets (S_{2i} or S_{2i+1}) is a 3D set, but does not contain any TSV in the bounding box (e.g., S_8 in Figure 38), S_i must select S_i^Z (= Z-cut). Our decision tree exploration is terminated at a unique solution satisfying one of above three conditions. The leaf nodes satisfy Conditions 1 or 2. A 3D clock tree can be obtained by traveling from the root node to the unique decision nodes.

The clock trees for all the unique sink sets are generated as follows. First, in abstract tree generation, we apply 2D-MMM algorithm [18] for a 2D sink set, or 3D-MMM algorithm [1] with TSV bound of 1 for a 3D sink set. We then perform clock tree embedding and buffering using classical DME method [90] or 3D slew-aware deferred-merge buffering and embedding (sDMBE) method [1] for zero skew under the Elmore delay model and satisfying the slew constraint.

5.2.4 Power Minimization with Decision Tree

Given a 3D sink set S_1 for the entire 3D design, our primary goal is to find the lowest power clock tree that connects all the sink nodes in S_1. Since either XY-cut or Z-cut is applied to S_1 (with the corresponding clock trees for sink subsets T_1^{XY} or T_1^{Z}), the minimum power value $P(S_1)$ is expressed as

$$P(S_1) = \min\{P(S_1^Z), P(S_1^{XY})\} \tag{16}$$

where $P(S_1^Z)$ and $P(S_1^{XY})$ are the minimum power values achieved with clock tress on subsets S_1^Z and S_1^{XY}.

It is possible that the clock design S_1^Z can be obtained directly by performing 3D-MMM algorithm [1] on S_1 with TSV bound of 1, selecting one TSV in the TSV array with lowest wiring cost, and then applying 3D clock tree embedding and buffering. The resulting clock tree S_1^Z uses one TSV, and $P(S_1^Z)$ can be calculated. However, low-power clock tree design for S_i^{XY} depends on the partitioning styles of its descendants.

Our DTCS methodology determines the power-optimal TSV array utilization based on a decision tree, where applying Z-cut vs. XY-cut at different partition levels leads to different power consumption and TSV array usage. Especially, applying an XY-cut on subset S_i ($i = 1, 2, \cdots, n$) results in two subsets S_{2i} and S_{2i+1}. The corresponding power estimation $P(S_i^{XY})$ is represented as follows:

$$P(S_i^{XY}) = P(S_{2i}) + P(S_{2i+1}) + P(S_{2i}, S_{2i+1}) \tag{17}$$

where $P(S_{2i})$ and $P(S_{2i+1})$ are clock power of subsets S_{2i} and S_{2i+1}, respectively. $P(S_{2i}, S_{2i+1})$ is the power of merging S_{2i} and S_{2i+1} that includes clock wires, buffers, and TSVs. Since the partitioning options for S_{2i} and S_{2i+1} are not determined yet, $P(S_i^{XY})$ can not be accurately determined at this point. Therefore, we need to first explore all possible cut orientations (XY-cut or Z-cut) for all descendant 3D subsets, which is done with our decision tree introduced in the previous Section 5.2.3. In addition, our clock tree construction algorithm to be discussed in Section 5.3.2 performs bottom-up traversal and accurately computes the power value for all nodes in the decision tree.

5.3 Decision-Tree-based Clock Synthesis Algorithms

5.3.1 Decision Tree Construction Algorithm

Our decision tree construction algorithm explores all the feasible solutions of TSV array utilization by recursively dividing a given sink set into two subsets in a top-down fashion. Given a sink set S_i, we first add a decision tree node d_i. For a node with Index i, the left and right child node indices are $2i$ and $2i + 1$, respectively. The root node has index 1. Second, we explore two candidate partitioning styles: Z-cut or XY-cut. We apply a Z-cut on S_i, obtain S_i^Z, and estimate the power consumption ($P(S_i^Z)$). Specifically, this is done by performing 3D-MMM algorithm [1] on S_i with the TSV bound of 1, where we select a TSV in the array with the lowest wiring cost in the bounding box of S_i^Z. Then, we embed the tree and insert buffers using sDMBE algorithm [1]. Next, we apply an XY-cut on S_i and obtain two subsets S_{2i} and S_{2i+1}. The $P(S_i^{XY})$ for 3D sink set can not be estimated at this point because the cut orientations for all descendants are not determined yet. If S_i is unique satisfying Conditions 1 or 2, the exploration is terminated, and node d_i obtains one solution only (i.e., S_i^Z or S_i^{XY} with estimated $P(S_i^Z)$ or $P(S_i^{XY})$ for 3D or 2D set, respectively); otherwise, we continue our top-down recursion on S_{2i} and S_{2i+1}.

5.3.2 Clock Tree Construction Algorithm

Our clock tree construction is to build an initial clock tree for power minimization using TSV arrays. If no TSV count constraint is specified, this initial tree becomes our final result. Otherwise, we perform clock tree refinement discussed in Section 5.3.3 to reduce the TSV count further down to satisfy the given TSV count constraint.

The input to our clock tree construction is a decision tree, where the cut orientations, clock trees, and their power values are determined for all leaf nodes. For internal nodes, only the clock trees and their power values are determined for Z-cuts. Thus, the goal is to determine the missing information: the clock tree and power value for XY-cut and the cut orientation for all internal nodes. We accomplish this goal by visiting each internal decision node d_i in a bottom-up fashion and determining the follows: (1) A clock tree and its power value for d_i for XY-cut (note that the clock tree and power are already available for the

Z-cut for d_i); (2) cut orientation for its children (not for d_i yet).

To compute the clock power for XY-cut ($P(S_i^{XY})$), we compare following 4 possible trees for d_i and choose the one with the lowest power (see Figure 39): (1) merge S_{2i}^Z and S_{2i+1}^Z; (2) merge S_{2i}^Z and S_{2i+1}^{XY}; (3) merge S_{2i}^{XY} and S_{2i+1}^Z; (4) merge S_{2i}^{XY} and S_{2i+1}^{XY}.

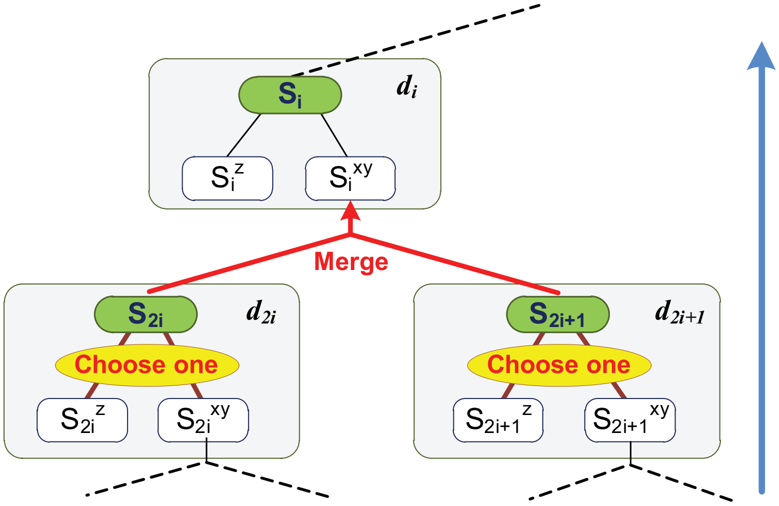


Figure 39: Bottom-up merging for node d_i, where we decide (1) clock tree and its power value for d_i for XY-cut ($= S_i^{XY}$), and (2) cut orientations for its children d_{2i} and d_{2i+1}.

$$P(S_i^{XY}) = \min \left\{ \begin{array}{c} P(S_{2i}^Z) + P(S_{2i+1}^Z) + P(S_{2i}^Z, S_{2i+1}^Z) \\ P(S_{2i}^Z) + P(S_{2i+1}^{XY}) + P(S_{2i}^Z, S_{2i+1}^{XY}) \\ P(S_{2i}^{XY}) + P(S_{2i+1}^Z) + P(S_{2i}^{XY}, S_{2i+1}^Z) \\ P(S_{2i}^{XY}) + P(S_{2i+1}^{XY}) + P(S_{2i}^{XY}, S_{2i+1}^{XY}) \end{array} \right\} \tag{18}$$

We select the merging combination that results in the lowest power for S_i^{XY} and assign the corresponding cut orientation decisions (Z+Z, Z+XY, XY+Z, or XY+XY) to the children d_{2i} and d_{2i+1}. The reason we consider these 4 merging options instead of simply propagating the minimum power bottom-up from the leaf nodes is due to the third term: $P(S_{2i}, S_{2i+1})$. This power overhead is caused by the wires, buffers, and TSVs used to merge the two children, and depends on the first two power terms $P(S_{2i})$ and $P(S_{2i+1})$. Thus, we build all possible merging and then pick the one with minimum power for accurate power evaluation.

Note that if we visit an internal node d_i whose children are leaf nodes, the cut orientation for the leaf nodes has been fixed already. In this case, the clock tree and XY-cut power of d_i are only based on a single combination. For all other internal nodes that do not have leaf nodes as children, the XY-cut power values are determined based on the 4 merging combinations shown in Equation (18). Especially, the root node d_1 selects the cut orientation (z_1) that results in the minimum power from $P(S_1^Z)$ and $P(S_1^{XY})$.

5.3.3 Clock Tree Refinement Algorithm

Note that our clock tree construction algorithm builds a 3D clock tree that uses no more TSVs than what is available in the TSV arrays in each die. However, if the TSV count is further bounded by a constant B, this constraint may not be satisfied. Therefore, we develop Clock Tree Refinement algorithm to reduce the TSV count below B while maintaining low power. The basic idea is to choose a subset of nodes in the decision tree and convert their cut orientation from XY to Z. Note that this conversion may lead to a TSV count reduction because an XY-cut node uses more-than-1 TSVs while a Z-cut node uses a single TSV. We develop a binary-integer-linear-programming (BILP)-based algorithm for this purpose to choose an optimal set of decision nodes.

The input to our BILP method is a decision tree with the cut orientation for all the nodes fixed and uses more TSVs than allowed. Each decision node d_i contains a sink set S_i, a total number of TSVs ($= t_i$) used in d_i for either S_i^{XY} or S_i^Z (which is 1 in this case), and the cost (c_i) if we convert the cut orientation from XY-cut to Z-cut. We first present our binary non-linear integer programming (BNLIP) as follows:

Minimize

$$\sum_{i \in D} c_i \times z_i \tag{19}$$

Subject to

$$\sum_{i \in D} (t_i - 1) \times z_i \geq T - B \tag{20}$$

$$z_i \times \sum_{j \in child(i)} z_j = 0, \text{ for } i, j \in D \tag{21}$$

$$z_i = \{0, 1\}, \text{ for } i \in D \tag{22}$$

The binary variable z_i represents the cut orientation for d_i with 0 for XY-cut and 1 for Z-cut.

We define D as the set of decision nodes that its XY-to-Z cut orientation conversion leads to a TSV count reduction. Note that for each $d_i \in D$, the $z_i = 0$ initially. The objective function Equation (19) is to minimize the total cost from cut orientation conversion. The cost c_i is the power overhead from the conversion and is expressed as:

$$c_i = P(S_i^Z) - P(S_i^{XY}) \tag{23}$$

where $P(S_i^Z)$ and $P(S_i^{XY})$ are the power of a Z-cut (i.e., using one TSV) or an XY-cut (i.e., using multiple TSVs) for S_i. Thus, our goal is to choose low-cost nodes for TSV count reduction.

Equation (20) ensures that the total TSV count should be no more than the given upper bound B, where T is the initial TSV count that exceeds the bound B. Once d_i is converted from XY-cut to Z-cut, S_i^Z is selected and one TSV is used instead. As a result, the total number of TSVs reduces by $t_i - 1$. Equation (21) shows that if node d_i is converted to Z-cut, the cut orientation of its children nodes z_j will not affect the overall power. This is because S_i^Z is a unique solution, and the decision tree is pruned at this node. Consequently, Equation (21) ensures that if $z_i = 1$, z_j for all descendant nodes are changed to 0.

Note that Equation (21) is not linear. For binary integer variables a and b_i, where $i = \{1, 2, .., n\}$, the quadratic constraint $a \times (b_1 + b_2 + ...b_n) = 0$ can be expressed as n linear constraints as $a + b_i \leq 1$ for $i = \{1, 2, .., n\}$. Thus, our binary integer linear programming (BILP) is formulated as Equations (19), (20), (24), and (22).

$$z_i + z_j \leq 1, \text{ for } i, j \in D, j \in child(i) \tag{24}$$

5.3.4 Extensions

Our methodology can be easily extended to handle more-than-two dies. A more-than-two-die 3D design can be decomposed into several pairs of two adjacent dies. Then, each two-die pair obtains a low-power 3D clock tree using our DTCS method. Finally, the 3D clock trees in those two-die pairs are connected through a single stacked TSV at the roots. Take a four-die stack (die-1 to die-4) as an example, we apply DTCS on die-1+die-2 and die-3+die-4, separately, and then connect the sources in the two 3D clock trees using a stacked TSV. As a result, the clock tree will use multiple TSVs mostly from two dies. We show 4-die stack clock tree designs in Section 5.4.3.

We extend the existing method 3D-MMM [1] to utilize TSV arrays. 3D-MMM is a top-down partition procedure, which assigns TSV bound for each subset based on the size and determines Z-cut primarily based on the given TSV bound in current partitioning. It chooses Z-cut if the current TSV bound is 1. This method inserts TSVs at any desired location. Thus, we modify it to handle TSV arrays in the following way. We first introduce additional constraints for Z-cut decision. Our major goal is to ensure that each 3D sink set will be assigned at least one TSV in the array within its bounding box under small routing overhead. A Z-cut will be applied to the current 3D sink set if any of the following conditions is satisfied: 1) if the bounding box of the 3D sink set contains only one TSV in the array; or 2) look ahead one more partitioning level down and check the availability of TSVs in the array. We apply an XY-cut at current partitioning and check for the bounding box of each subset. If a subset contains 3D sinks but has no TSV in the bounding box, we apply a Z-cut at current iteration. Note that if the available TSV is outside the bounding box of a 3D sink set, a routing detour will be added. Thus, we use bounding box of a 3D sink set to determine the availability of the TSVs in the array.

5.4 Experimental Results

5.4.1 Simulation Setting

In our experiments, we compare the following four algorithms to demonstrate the effectiveness of our DTCS method:

- **ALG-D**: our decision-tree-based clock synthesis (DTCS) algorithm that is specific for 3D clock routing with TSV arrays.
- **ALG-M**: extension on the existing method 3D-MMM [1] to utilize TSV arrays, which is described in Section 5.3.4.
- **ALG-F**: existing method 3D-MMM [1] that freely inserts TSVs at desired positions without the TSV array constraint.
- **ALG-X**: results in the optimal design. It exhaustively enumerates all feasible solutions of TSV array utilization and selects the tree with minimum power satisfying the

TSV bound.

We implemented the above four algorithms in C++/STL and performed experiments on a 64-bit Linux server with Intel 2.5GHz CPU. 3D clock trees operate at 1GHz frequency and 1.1V supply voltage. We report clock power, slew, and skew from SPICE simulation, where skew and slew are constrained below 30ps and 100ps, respectively. Our DTCS (=ALG-D) solves the BILP problem using MOSEK. We use 45nm PTM model. The TSV parasitic resistance and capacitance are $35m\Omega$ and $15f\text{F}^2$, respectively.

We performed verifications on various types of benchmark circuits (see Table 9): 12 gate-level circuits with 1K to 17.6K clock sinks, where each of the ckts contains both 2-die and 4-die designs; and 4 block-level 2-die circuits (ckts 7-10).

Table 9: Benchmark designs. Footprint area is in $mm \times mm$.

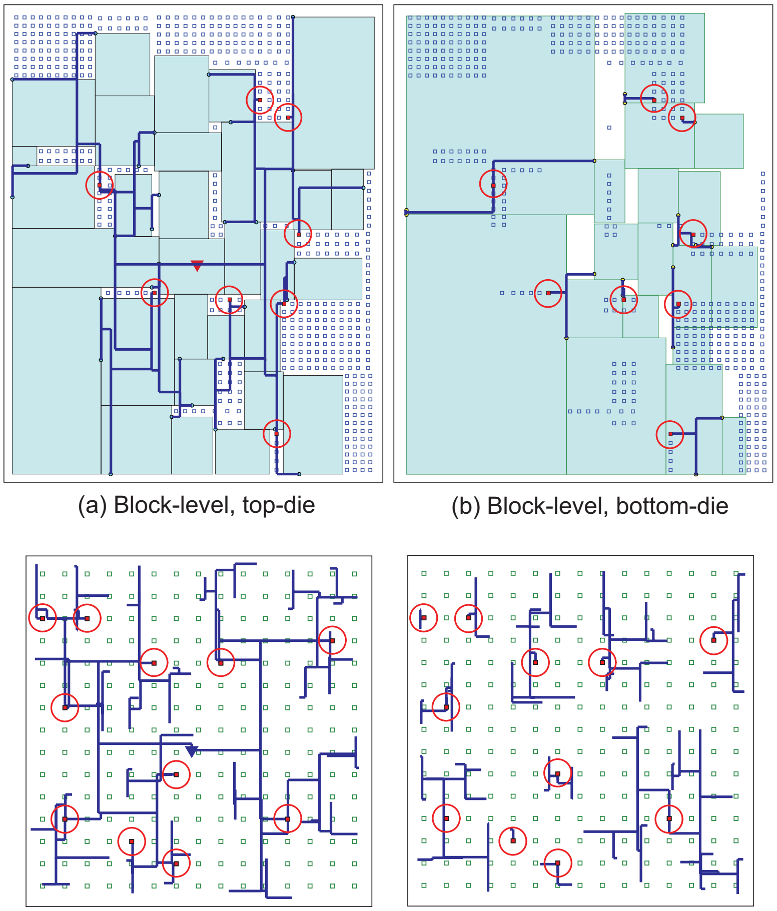
gate-level							
		2-Die			4-Die		
ckt	#sinks	footp area	TSV array	TSV bnd	area	TSV array	TSV bnd
ckt1	1089	3.7 × 3.7	32 × 32	120	2.6 × 2.6	20 × 20	130
ckt2	3204	5.3 × 5.4	36 × 36	250	3.8 × 3.7	28 × 28	180
ckt3	12404	12.3× 12.1	52 × 52	271	8.7 × 8.6	46 × 46	544
ckt4	1090	4.0 × 3.7	32 × 32	70	2.7 × 2.6	24 × 24	80
ckt5	12340	6.1 × 6.1	48 × 48	331	4.3 × 4.3	32 × 32	180
ckt6	17616	7.0 × 7.0	52 × 52	320	4.9 × 4.9	42 × 42	400
block-level, 2-Die only							
ckt	#2D blocks	footp area	#TSV blocks	TSV bnd			
ckt7	33	7.9×8.7	16	5	-	-	-
ckt8	49	3.9×4.8	39	5	-	-	-
ckt9	300	3.9×5.0	161	35	-	-	-
ckt10	51	5.7×6.6	30	7	-	-	-

These benchmark circuits come from IWLS05, MCNC, GSRC, and ISPD09. For ckts 4-10, we perform 3D placement [63] for gate-level designs and 3D floorplanning for block-level designs. Both are Cadence Encounter-based 3D physical design tool chains. For ckt 1 (from ISPD09) and ckts 2-3 (from GSRC), since they only include the information of clock sinks for 2D ICs, we duplicate the circuits, scale the footprint area by $1/\sqrt{2}$ and $1/\sqrt{4}$ for 2-die and 4-die, respectively, then randomly assign sinks on different dies and insert TSV

[2] Our DTCS takes into account the TSV parasitics in power evaluation and is efficient for any given TSV parasitics.

arrays accordingly. The TSV bound constraint for each die is set as 5% to 20% of total TSV count in the arrays per die for gate-level and 10% of 2D block count for block-level design.

Two 3D clock trees generated by our ALG-D for the TSV arrays are shown in Figure 40, where Figures 40(a) and 40(b) are for block-level ckt8 using 8 TSVs, and Figures 40(c) and 40(d) are for gate-level circuit with 121 sinks in two-die stack with 11 TSVs. The top die contains a complete 2D tree and the bottom die contains many subtrees. The utilized clock TSVs are highlighted by red circles.


(a) Block-level, top-die

(b) Block-level, bottom-die

(c) Gate-level, top-die

(d) Gate-level, bottom-die

Figure 40: Clock trees generated by our ALG-D using TSV arrays. We show 3D clock trees for block-level ckt8 ((a) and (b)) and a gate-level ckt ((c) and (d)) in top- and bottom-die, respectively. TSV arrays are denoted as squares. Clock TSVs are shown in red circles.

5.4.2 Comparison with ALG-X

Comparisons between ALG-X and our ALG-D are shown in Table 10, which includes power (mW) and runtime (s) for a 2-die circuit with 121 sinks under TSV bound of 7. The TSV array size increase from 4x4 to 7x7. Our ALG-D efficiently finds close-to-optimal solutions in short runtime for both designs with and without TSV bound.

Table 10: Comparison between ALG-X and our ALG-D in power (mW) and runtime (s).

	No TSV Bound				With TSV Bound			
TSV	Power		Runtime		Power		Runtime	
array	ALG X	**ALG D**	ALG X	**ALG D**	ALG X	**ALG D**	ALG X	**ALG D**
4x4	64.28	**64.34**	20	**0.12**	67.03	**67.27**	20	**0.16**
5x5	64.88	**64.93**	130	**0.11**	65.54	**65.54**	130	**0.17**
6x6	63.82	**63.85**	2357	**0.14**	64.53	**64.53**	2357	**0.18**
7x7	62.31	**62.31**	79813	**0.15**	63.48	**63.57**	79813	**0.21**
Inc.		**0.05%**				**0.13%**		

First, our ALG-D results in no more than 0.4% power increase compared with the optimal design obtained in ALG-X. This demonstrates the effectiveness of our clock tree construction and refinement in DTCS. Second, our ALG-D finishes routing within 0.3 seconds. But, ALG-X may cost more than tens of hours depending on the TSV array size. The runtime of ALG-X is unaffordable for two reasons. The solution space tremendously expands in larger TSV array: ALG-X synthesizes 677 clock trees in 20 seconds for 4×4 TSV array, but 2648145 trees in 79813 seconds (>22 hours) for the 7×7 TSV array. Next, a larger circuit with more than 10k sinks requires longer runtime for each run of 3D clock routing. Lastly, as expected, we observe that our clock tree with a TSV bound utilizes fewer TSVs in the TSV array and consumes more power and longer runtime than the design with no TSV bound constraint. This extra runtime comes from our BILP-based clock tree refinement.

5.4.3 Comparison with Related Work

The comparisons of our ALG-D with ALG-F and ALG-M for gate-level and block-level circuits are shown in Table 11. Results are obtained under three TSV bound constraints, i.e., single, no TSV bound, and with a bound. Detailed clock routing results are also presented for the designs with no TSV bound.

Limitations in ALG-M: As discussed early, TSV array utilization significantly affects the clock power consumption. The straightforward extension for TSV arrays (ALG-M) cannot efficiently reduce clock power when using many TSVs. When TSVs are irregularly inserted at any desired location (with no TSV array limitation), ALG-F [1] achieves 12% average power reduction compared with using single TSV in a 3D clock network. However, to support the TSV array design style, the ALG-M, a straightforward extension on existing method [1], did not efficiently reduce the power by using multiple TSVs. Instead, it may waste tens to hundreds of TSVs. The major reason is that the existing algorithm does not take into account the TSV array limitations, where using many TSVs may result in detour and extra TSV parasitic capacitance.

Our ALG-D versus ALG-M: Detailed results are presented in Table 11, which includes wirelength, TSV count, buffer count, and power consumption for the comparisons between ALG-M and our ALG-D. Our observations are as follows. First, in the design with no TSV bound, our ALG-D efficiently minimizes clock power by utilizing TSV arrays. Compared with ALG-M, our ALG-D achieves 13.5%, 11.3%, 15.7% reduction in clock power, wirelength, and buffer count, respectively. In addition, our ALG-D uses 55.1% fewer TSVs than ALG-M. Second, in the design with bounded TSVs, our ALG-D achieves 9.1% power reduction on average. Third, clock skew is well controlled below 30 ps. The ALG-D runtime of the designs is in the range of 5 to 40 seconds based on the circuit size.

Our ALG-D versus ALG-F: Note that ALG-F [1] freely inserts TSVs at any desired position. Our ALG-D results in comparable or even lower power than ALG-F. In the design with no TSV bound, ALG-F and our ALG-D can achieve 12% and 11% average power reduction compared with the single-TSV solution, respectively. With a TSV bound, our ALG-D generates clock designs with 0.90 average power ratio, whereas, ALG-F obtains 0.94 average power ratio.

5.5 Summary

In this chapter, for the first time, we studied low-power 3D clock design with TSV arrays. The TSV array design style is essential for reliable 3D ICs, but it significantly affects the power efficiency due to the constrained TSV locations. We presented a novel methodology, so called decision-tree-based clock synthesis (DTCS), to generate high-quality and low-power 3D clock trees by efficiently exploring the entire solution space for the best TSV array utilization. Our DTCS algorithm consists of decision tree construction, clock tree construction, and clock tree refinement. We demonstrated the effectiveness of our DTCS method for both chip-scale gate-level and block-level 3D IC designs. The following conclusions have been drawn. First, our DTCS algorithm obtains close-to-optimal solutions in short runtime, compared with the method of exhaustive searching TSV utilizations. Second, a straightforward extension on the existing algorithm for TSV arrays can not generate low-power 3D clock network, but waste many TSVs. Third, compared with the extension of existing algorithm, our DTCS algorithm achieves 13.5% and 9.1% power reduction in various given TSV bounds, uses 55.1% fewer TSVs, and obtains 11.3% shorter wirelength and 15.7% fewer buffers on average.

Table 11: Comparison among ALG-F [1], ALG-M, and our ALG-D with no TSV bound and with TSV bound. Detailed results of power (mW), wirelength (μm), TSV count, buffer count, and skew (ps) in no TSV bound designs are shown.

	Power	With No TSV Bound											With TSV Bound				
	single	Power			#TSVs		WireLength		#BUFs		skew		Power			#TSVs	
ckt_#die	TSV	alg-F	alg-M	**alg-D**	alg-M	**alg-D**	alg-M	**alg-D**	alg-M	**alg-D**	alg-M	**alg-D**	alg-F	alg-M	**alg-D**	alg-M	**alg-D**
ckt1_2die	183.5	160.0	178.9	**155.2**	454	**181**	273769	**246760**	1279	**1105**	26.3	**28.0**	169.0	175.5	**158.8**	120	**120**
ckt2_2die	609.5	553.1	595.3	**523.0**	818	**318**	748353	**651223**	4164	**3791**	28.9	**28.1**	581.6	580.0	**532.0**	250	**250**
ckt3_2die	1445.1	1273.4	1518.6	**1304.8**	1378	**526**	2348205	**1974101**	10027	**9034**	28.9	**26.1**	1372.9	1499.9	**1331.4**	271	**271**
ckt4_2die	189.7	176.5	192.1	**170.8**	220	**88**	388123	**361190**	732	**580**	25.8	**23.1**	188.0	187.6	**171.2**	70	**70**
ckt5_2die	539.4	495.3	554.7	**495.8**	662	**447**	1369811	**1252539**	1970	**1626**	24.9	**21.5**	522.5	552.7	**499.2**	331	**330**
ckt6_2die	970.5	893.3	1087.5	**877.2**	1908	**434**	2111595	**1758220**	3894	**3073**	25.8	**24.1**	924.6	1007.8	**882.7**	320	**320**
ckt1_4die	178.3	162.1	175.2	**151.4**	358	**159**	271870	**241801**	1212	**1041**	26.2	**18.7**	167.6	169.6	**153.7**	129	**130**
ckt2_4die	560.2	488.6	538.8	**466.0**	828	**360**	674576	**583082**	3712	**3298**	26.2	**9.7**	506.4	547.8	**485.1**	180	**180**
ckt3_4die	1413.7	1227.7	1386.6	**1217.6**	1664	**736**	2156004	**1884300**	9552	**8614**	28.6	**25.0**	1295.3	1380.3	**1231.4**	544	**544**
ckt4_4die	194.2	172.9	201.4	**177.6**	327	**105**	414419	**376671**	701	**592**	22.8	**22.1**	183.3	191.7	**179.0**	80	**80**
ckt5_4die	530.7	493.2	596.9	**504.3**	584	**259**	1493940	**1288392**	2085	**1659**	25.1	**12.6**	513.0	552.7	**506.8**	173	**177**
ckt6_4die	987.1	905.6	1100.1	**899.8**	2679	**579**	2070177	**1767473**	3879	**3111**	24.9	**21.2**	940.3	1002.4	**906.3**	392	**400**
ckt7_blk	30.5	24.4	29.4	**27.6**	11	**8**	80410	**76182**	96	**89**	11.8	**18.8**	26.5	30.1	**28.5**	5	**5**
ckt8_blk	19.0	17.1	19.9	**16.2**	11	**8**	53340	**45546**	67	**49**	10.5	**5.4**	18.4	18.8	**17.5**	5	**5**
ckt9_blk	58.2	46.3	55.7	**51.2**	94	**47**	142370	**135997**	189	**168**	9.3	**10.5**	51.8	54.6	**51.4**	35	**35**
ckt10_blk	26.5	21.0	28.1	**23.8**	20	**9**	76091	**67653**	92	**70**	22.1	**22.1**	24.8	27.9	**24.4**	7	**7**
Avg. Ratio	1.00	0.88	1.02	**0.89**	1.00	**0.45**	1.00	**0.89**	1.00	**0.84**			0.94	0.99	**0.90**		
% Improv. of alg-D w.r.t. alg-M:				**13.5%**		**55.1%**		**11.3%**		**15.7%**					**9.1%**		

CHAPTER VI

THREE-DIMENSIONAL POWER NETWORK ANALYSIS FOR ELECTRO-MIGRATION RELIABILITY

Power-delivery network (PDN) design has become a challenging task in integrated circuit (IC) design. Since the supply voltage scales slower than transistors and interconnects do, the current density has been rapidly increasing. This increased current density along with the high temperature accelerates the degradation of transistors and wires and shortens the lifetime of both devices and wires.

Power-delivery networks provide supply voltage to all devices in the entire three-dimensional (3D) stack. The inter-die power-delivery interconnects, formed by power/ground (P/G) through-silicon vias (TSVs) or micro-bumps, are unique components in 3D power grids. Because these vertical connections carry large amounts of current, the 3D power networks may suffer from electro-migration (EM) degradation. Therefore, the detailed and accurate analysis on the 3D PDN is important to predict the performance and to improve the power integrity.

This chapter focuses on studying the current-density distribution inside TSVs and the impact of current crowding on power integrity. A small cross section of the global 3D PDN is illustrated in Figure 41. Two dies are bonded face-to-back and are connected using via-last TSVs. The voltage is supplied from the package through the controlled-collapse chip connection (C4). In the bottom die, the current is delivered directly to Metal 10 and Metal 9. However, in the top die, the current is delivered to Metal 10 and Metal 9 through TSVs. Both the intermediate and local sections of the PDN are connected using local vias to the global PDN. This generic structure is used for both isolated TSV modeling and large-scale 3D PDN modeling.

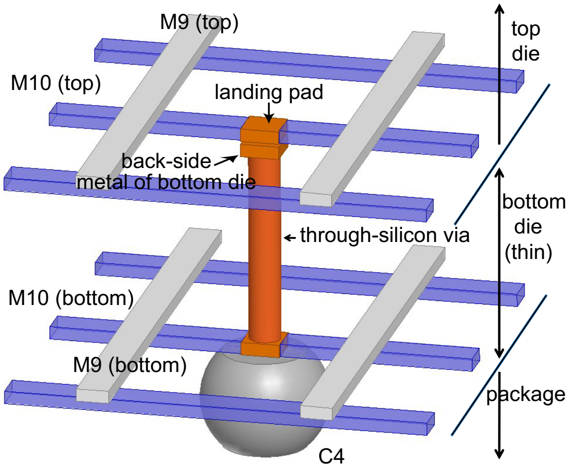


Figure 41: 3D connection in a global power-delivery network.

6.1 Current Crowding in 3D ICs

6.1.1 Current-Density Distribution inside a TSV

The test case used to investigate the current-density distribution inside a TSV is shown in Figure 42. This corner case is chosen specifically to study a highly asymmetric current distribution, which consists of the following components: (1) One TSV with a 5μm diameter and a 30μm height; (2) two 6μm×6μm landing pads; (3) two 2μm-wide power wires on the top; and (4) one 6μm-wide power wire on the bottom.

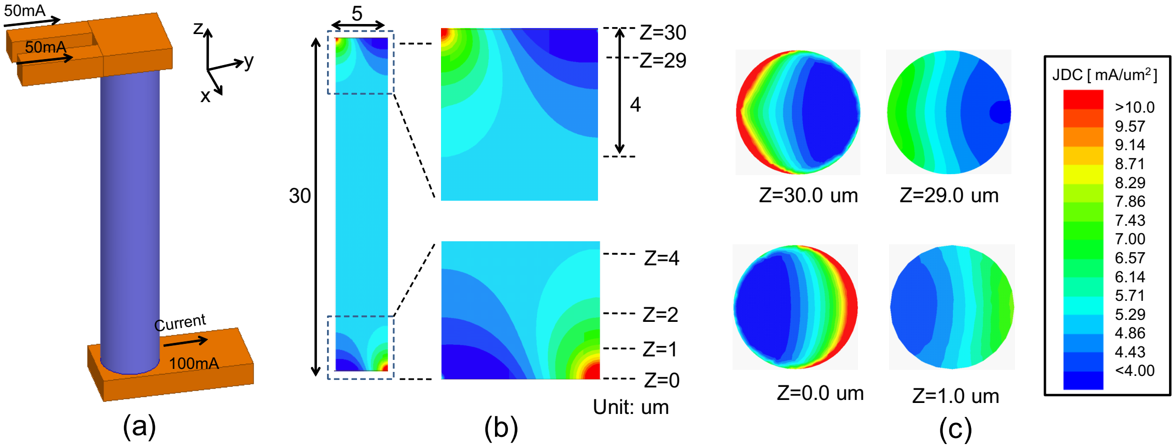


Figure 42: Current crowding in the test case of a TSV and power wires (a). The current-density distribution is shown in a ZY plane (b) and in top-down XY planes (c).

In Figure 42 (a), the thickness of the power wire is 2μm. The copper resistivity is 18Ω·nm. Two current sources are inserted at the top-left corner, each sourcing a 50mA current. A current sink is defined at the bottom-right corner. This test case constrains the direction of current flow and is used to investigate the current-density distribution in the TSV. ANSYS Q3D [91], a finite element tool, is used to simulate the current-density distribution and the voltage drop.

The magnitude of current density is plotted for several cross sections in Figures 42(b) and 42(c). In Figure 42(b), a large portion of current from the power wires dives into the top-left TSV edge and flows out at the bottom-right edge. Compared with the average current density inside the TSV, which is 5.1mA/μm^2, the edge current density is approximately 10mA/μm^2. For the current-density distribution on the ZY plane, a significant current crowding is observed. This crowding occurs at 4μm into the TSV from both the top and the bottom interfaces along the Z-axis. In the center region of the TSV, where Z is between 4μm and 26μm, the current is uniformly distributed inside the TSV. Current-density distributions on the XY planes are depicted in Figure 42(c), where Z is 30.0μm, 29.0μm, 1.0μm, and 0.0μm. Most of the current is concentrated at the connection between the power wires and the TSV.

6.1.2 TSV-Diameter-to-Wire-Thickness Ratio

The magnitude of current crowding depends on the ratio of the TSV diameter to the wire thickness. Current-density distributions under different wire thickness are illustrated in Figure 43, where the TSV diameter is fixed at 5μm, and the wire thickness is changed from 1.0μm to 3.0μm.

In the case of the 3um-thick wires, a significant amount of current is shunted over the power wire instead of concentrating at the edge. This phenomenon is due to the low-resistive path in the thick wire. If two designs have the same TSV-diameter-to-wire-thickness ratio, the current-density distributions will be the same. For example, the current density of a design using 5.0μm-wide TSVs and 1.0μm-thick wires will be similar to the current density of a design using 10.0μm-wide TSVs and 2.0μm-thick wires.

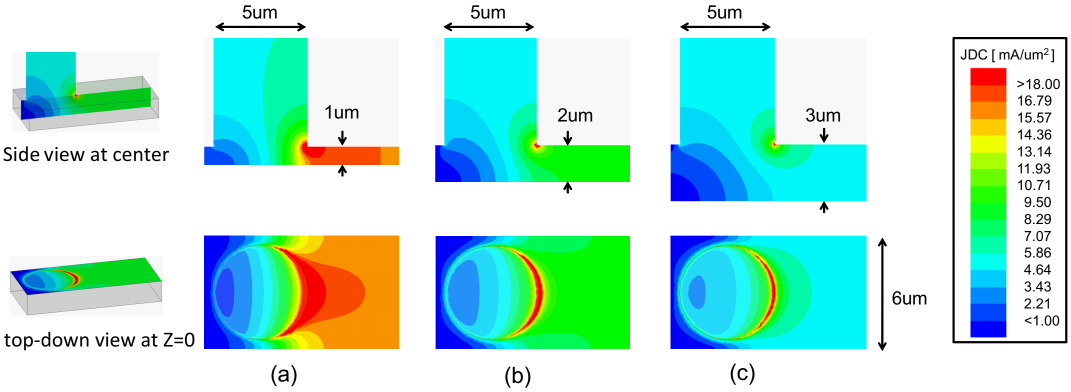


Figure 43: The ratio of the TSV diameter to the wire thickness affects the current crowding at the connection corner. The TSV diameter is set to 5.0µm, and the power thickness is 1.0µm (a), 2.0µm (b), and 3.0µm (c).

The maximum current density (J_{max}) inside the TSV due to current crowding is shown in Table 12, where the TSV diameter changes from 16.0µm to 2.0µm, and the wire thickness is held constant at 2.0µm. When the TSV diameter is 16.0µm, the maximum current density is more than 10 times larger than the average value (J_{avg}). When the TSV diameter is 2.0µm, however, the maximum current density is twice of the average value. Therefore, a high maximum current density can occur at the edge of the TSV that has a large diameter.

Table 12: Impact of the TSV diameter on the current crowding. The TSV delivers 100mA current, and the wire thickness is 2.0µm.

	Case 1	Case 2	Case 3	Case 4	Case 5
TSV diameter (µm)	16	8	5	4	2
TSV height (µm)	48	48	30	24	12
Power wire length (µm)	18	10	6	5	3
J_{avg} (mA/µm^2)	0.5	2.0	5.1	8.0	31.8
J_{max} (mA/µm^2)	5.5	10.4	19.2	25.8	62.0
J_{max}/J_{avg}	11.1	5.2	3.8	3.2	2.0
TSV diameter: wire thickness	8:1	4:1	2.5:1	2:1	1:1

6.1.3 Impact of Current Crowding on IR Drop

The current crowding inside the TSV changes the effective resistance of the TSV as well as the voltage drop across the TSV. Because the spreading resistance [92] is caused by the nonparallel current between two spatially separated contacts, the effective resistance of the TSV due to current crowding is larger than the value obtained using $R_0 = \rho \times l / A$, where ρ is the resistivity, l is the length, and A is the cross-sectional area of the TSV.

ANSYS Q3D extractor is used to simulate the voltage drop across the TSV. In these simulations, the TSV dimensions are held constant, and the wire thickness increases from 1.0µm to 3.0µm. The resulting voltage drop through the TSV is shown in Table 13.

Table 13: Impact of current crowding on voltage drop through a TSV. The thickness of power wire varies from 1.0µm to 3.0µm.

Wire thickness (µm)	1.0	2.0	3.0
Voltage drop w/ current crowding (mV)	3.33	3.11	3.02
Voltage drop w/o current crowding (mV)	2.75	2.75	2.75
Increase by current crowding (%)	21.1	13.1	9.8

For a 100mA current, the voltage drop through R_0 is IR_0=2.75mV, which is not affected by the wire thickness. However, since current crowding is sensitive to the wire thickness, as the wire thickness increases from 1.0µm to 3.0µm, the voltage drop decreases from 3.33mV to 3.02mV, which corresponds to 21.1% to 9.8% greater voltage drop than the calculated value.

6.1.4 Interface of Power Wires and TSVs

The current-density gradient occurs not only at the edge of the TSV but also at the connections between the power wires and the TSV landing pad as shown in Figure 44. Before connecting to the landing pad, the current density inside power wires is relatively uniform. In the transition region, the current concentrates toward the nearest interface between the TSV and the landing pad.

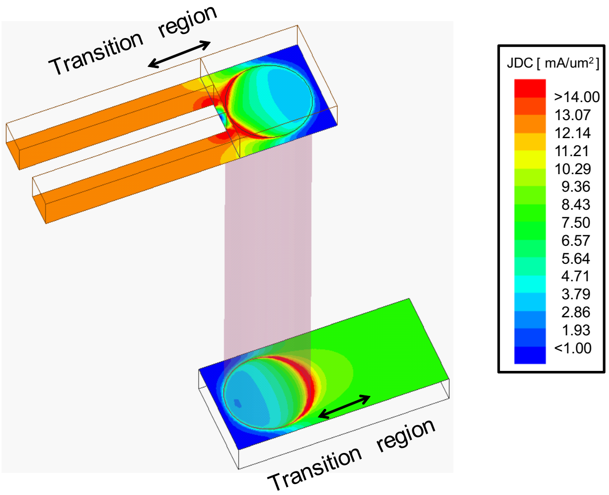


Figure 44: Current crowding in the transition region between power wires and TSVs.

6.2 TSV Current Crowding Model

In traditional PDN modeling, power wire segments and TSVs are modeled as lumped resistors. This traditional model can only represent uniform current densities, which is insufficient to accurately capture non-uniform current distributions caused by current crowding. Likewise, modeling the TSV as a single resistor is also insufficient to accurately calculate the voltage drop that is related to the spreading resistance and depends on current distributions.

This section describes a TSV model that allows non-uniform current densities within a TSV and its transition regions. The proposed model can be easily integrated into netlists for chip-scale PDN analysis and is simple enough that runtime remains reasonable. An illustration of the TSV model is shown in Figure 45.

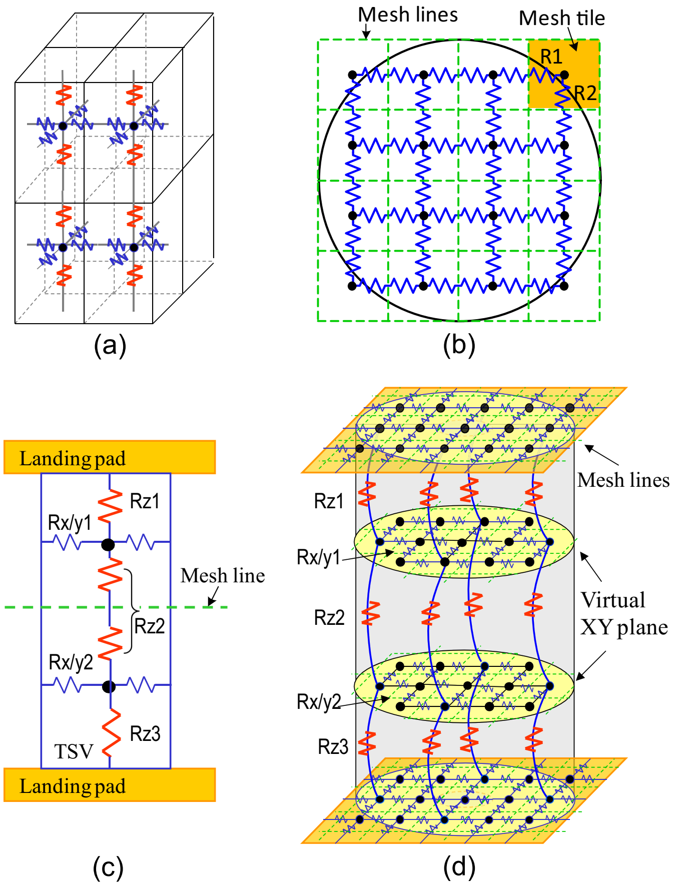


Figure 45: The proposed TSV modeling approach. Basic rectangular box after 3D meshing (a); XY-mesh and partially overlapped mesh tiles (b); side view (c); 3D view of the network (d).

6.2.1 3D Resistance Network for TSV Modeling

A TSV is modeled as rectangular mesh boxes as depicted in Figure 45(a), where each mesh box consists of six resistors: east, south, west, north, up, and down. These rectangular mesh boxes are connected to the neighboring boxes at the center connecting points. The 3D mesh structure of a TSV is generated as follows: (1) Z-mesh: The TSV is divided into multiple short cylinders with the same diameter but various thicknesses and (2) XY-mesh: Each short cylinder is then meshed into a 2D resistance network on a virtual XY plane, which is located at the center of each cylinder.

Virtual XY planes are created by partitioning the TSV along the Z-axis. The Z locations of these XY planes, referred to as the Z-mesh, are determined by the current gradient on the ZY plane. The region with a large current crowding contains more cylinders than the region with a uniform current density. Specifically, the Z-mesh size is fine near both top and bottom landing pads and is coarse in the middle of the TSV.

A side view of the 3D resistance network is shown in Figure 45(c), where two virtual XY planes are generated. The resulting model is a non-uniform 3D resistance network consisting of two types of resistors: (1) The resistors along the Z-axis (R_{z1}, R_{z2}, and R_{z3} in Figures 45(c) and 45(d)) that are connected to the neighboring virtual XY planes and (2) the resistors in virtual XY planes ($R_{x/y1}$ and $R_{x/y2}$ in Figures 45(c) and 45(d)).

If a mesh tile is completely covered by the real TSV shape, R_z and $R_{x/y}$ are directly obtained referring to the size of XY-mesh and Z-mesh. However, around the TSV boundary, mesh tiles partially overlap with the real shape as shown in Figure 45(b). For this case, the overlap area is calculated as the cross-sectional area for R_z calculation, and the effective length along the X-axis and Y-axis is then obtained for R_1 and R_2 calculation.

A schematic of the proposed modeling approach is depicted in Figure 45(d). Most virtual planes and resistors are not shown for readability. The non-uniform Z-mesh used in the model is a trade-off between complexity and accuracy. The 30µm-high TSV is vertically partitioned at 0.1µm, 0.4µm, 0.9µm, 2.0µm, 5.0µm, 16.0µm, 27.0µm, 28.9µm, 29.4µm, 29.7µm, 29.9µm, and 30.0µm. Three different XY-mesh sizes are implemented for comparisons: 0.25µm, 0.5µm, and 1.0µm.

6.2.2 Modeling of Transition Region

A transition region is defined as the connection area between a power wire and a TSV landing pad. The meshing result on the transition region is shown in Figure 46, where a non-uniform current gradient can occur.

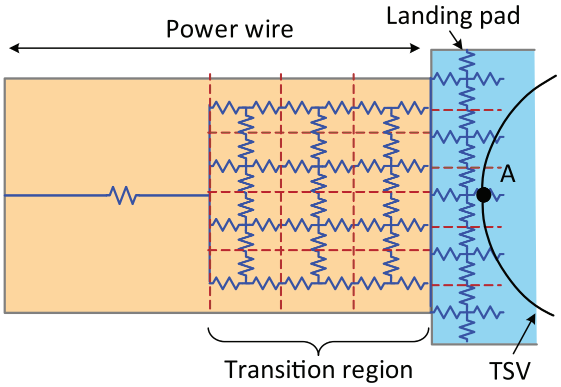


Figure 46: Meshing on the transition region.

Although the total current flowing into the transition region is equal to that out of the region, the local current density at the landing pad depends on the meshing structure. Without meshing the transition region, the entire current would entirely flow into Point A, which results in a large but incorrect current at the edge of the TSV. By meshing the transition region, the current spreads evenly along the power wire and then flows into the landing pad and the TSV edge, which results in a high accuracy. A transition region approximately 6.0µm long is found to be long enough.

6.2.3 Modeling Accuracy

Detailed comparisons between ANSYS Q3D and a power simulator (PSIM) are shown in Figure 47, where PSIM models TSVs using the proposed approach, and the XY-mesh size is 0.25µm. For the PSIM results, the current in each mesh tile is extracted and divided by the effective area. For the Q3D results, the current gradient is simulated by running the internal mesh generator and solver, and the current values are mapped into the mesh tile structure.

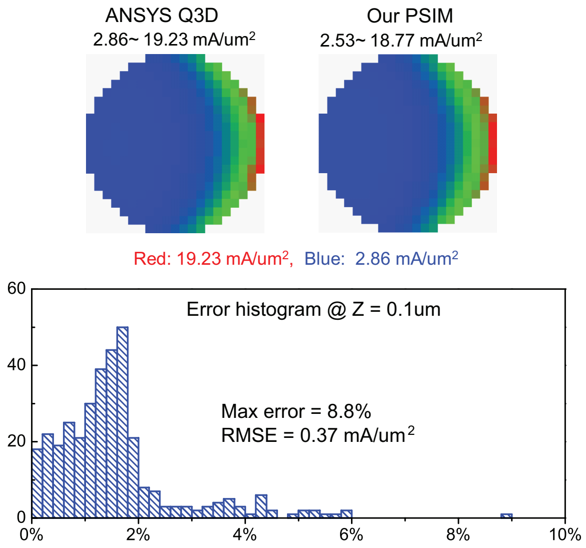


Figure 47: Current density distributions and the error histogram of ANSYS Q3D and the proposed TSV modeling approach in PSIM at Z=0.1µm. The error in each tile is the absolute difference between Q3D and PSIM.

The current-density distributions obtained from Q3D and PSIM are plotted in the top half of Figure 47, where the virtual plane locates at Z=0.1µm. The error histogram of the current density between Q3D and PSIM is shown at the bottom of Figure 47, where the error for each tile is defined as the absolute difference between Q3D and PSIM. This comparison is for the closest virtual XY plane to the landing pad, where the largest current crowding is observed.

PSIM has a very good accuracy compared with Q3D. The relative error of PSIM for each mesh tile is less than 10%, and most of the errors are within 5%. The root-mean-square error (RMSE) is expressed as follows:

$$\text{RMSE} = \sqrt{(\sum_{i=1 \text{ to } n} (J_i^{Q3D} - J_i^{PSIM})^2)/n} \ , \tag{25}$$

where i is the i^{th} tile, and n is the total number of tiles. The RMSE of the proposed method is 0.36mA/µm^2. The voltage drop of PSIM is 3.07mV, which is 0.33% different from the Q3D result.

The differences between Q3D and PSIM are mainly due to the mesh structure. The proposed model uses low-density orthogonal meshing boxes for simplicity, whereas Q3D supports sophisticated meshing structures, e.g., triangular and tetrahedral shapes. However, the simulation time of PSIM is less than one second; whereas Q3D takes up to one hour. These comparisons demonstrate that the proposed modeling approach has the potential to analyze the chip-scale power integrity with a reasonable accuracy and an acceptable runtime.

6.2.4 Impact of XY-Mesh Size

The impact of XY-mesh size on the accuracy of the proposed model in terms of current density and voltage drop is shown in Table 14. The Z-mesh size is held constant, and the XY-mesh size is increased from 0.25μm, 0.5μm, to 1.0μm.

Using larger meshing tiles, the RMSE of the current density increases from 0.25mA/μm^2 to 0.55mA/μm^2, which is equal to 4.9% to 10.7% of the average current density. To report the maximum current density in Q3D for a given mesh size, the Q3D simulation result is mapped into each mesh tile. Thus, the maximum current density of Q3D reduces with different mesh sizes in Table 14 as well. The error of the maximum current density increases from 2.1% to 27.9%, and the voltage drop error increases from 0.3% to 3.9%.

Table 14: Impact of the XY-mesh size on the current density (mA/μm^2) and the voltage drop (mV).

mesh			Max. Current density			Voltage drop		
(μm)	#tiles	RMSE	Q3D	PSIM	err (%)	Q3D	PSIM	err (%)
0.25	4641	0.25	19.2	18.8	-2.1	3.1	3.10	0.3
0.5	1313	0.34	18.0	20.8	15.6	3.1	3.09	0.7
1.0	325	0.55	12.2	15.6	27.9	3.1	2.99	3.9
none	1	–	19.2	5.1	73.4	3.1	2.75	11.3

The cost of using finer size is that the total number of mesh tiles increases from 325 to 4641. Simulation results using a single resistor are also shown in the table, which results in average current density of 5.1mA/μm^2 (73.4% smaller than the maximum current density from Q3D) and lower voltage drop of 2.75mV.

6.3 Chip-Scale 3D PDN Analysis

6.3.1 Chip-Scale PDN Circuit Model

A circuit model of a partial 3D PDN is illustrated in Figure 48. This model is developed to analyze global PDNs that have high current densities and contain TSV connections. Both power wire segments and local vias are represented as lumped resistors. The 3D power connection is modeled using the proposed approach, which includes TSVs and transition regions. An ideal voltage of 1V is supplied from the C4. The current sinks in each die are located at the intersections of the power grids. The power wires are 2.0µm thick and 5.0µm wide. The TSVs are 30.0µm high and 5.0µm wide. The size of landing pads is 6.0µm×6.0µm.

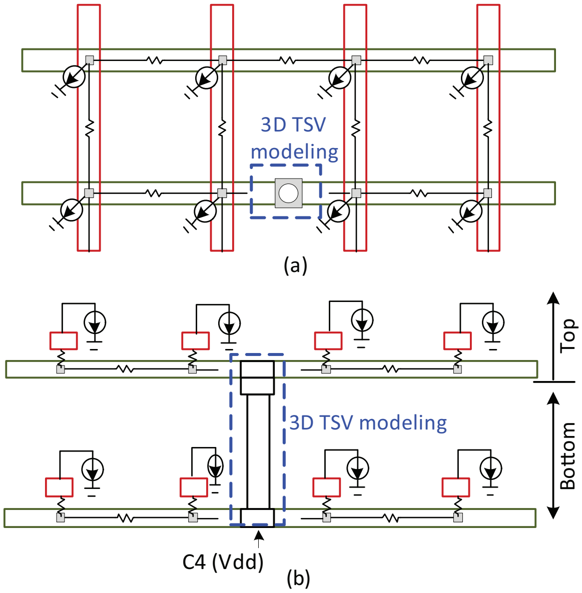


Figure 48: A circuit model for a two-die TSV-based PDN using the proposed 3D TSV modeling approach in top-down view (a) and side view (b).

6.3.2 Simulation Results

Two voltage-drop maps and one power map of a global PDN are shown in Figure 49. The footprint area is 1.4mm×1.4mm. Each die has 16×16 power wires and a 15µm-thick power ring around the boundary. TSVs and C4s are aligned in the bottom die, which are enlarged as white blocks for readability. Current sinks are represented as black boxes at the intersections of power wires.

The power map in the bottom die is shown in Figure 49(c). Power maps in both top die and bottom die have a cool spot in the bottom-left corner and a hot spot in the top-right corner. In the center of each die, another two narrow cool spots are placed on the left and right. These power maps result in different current-density patterns surrounding the

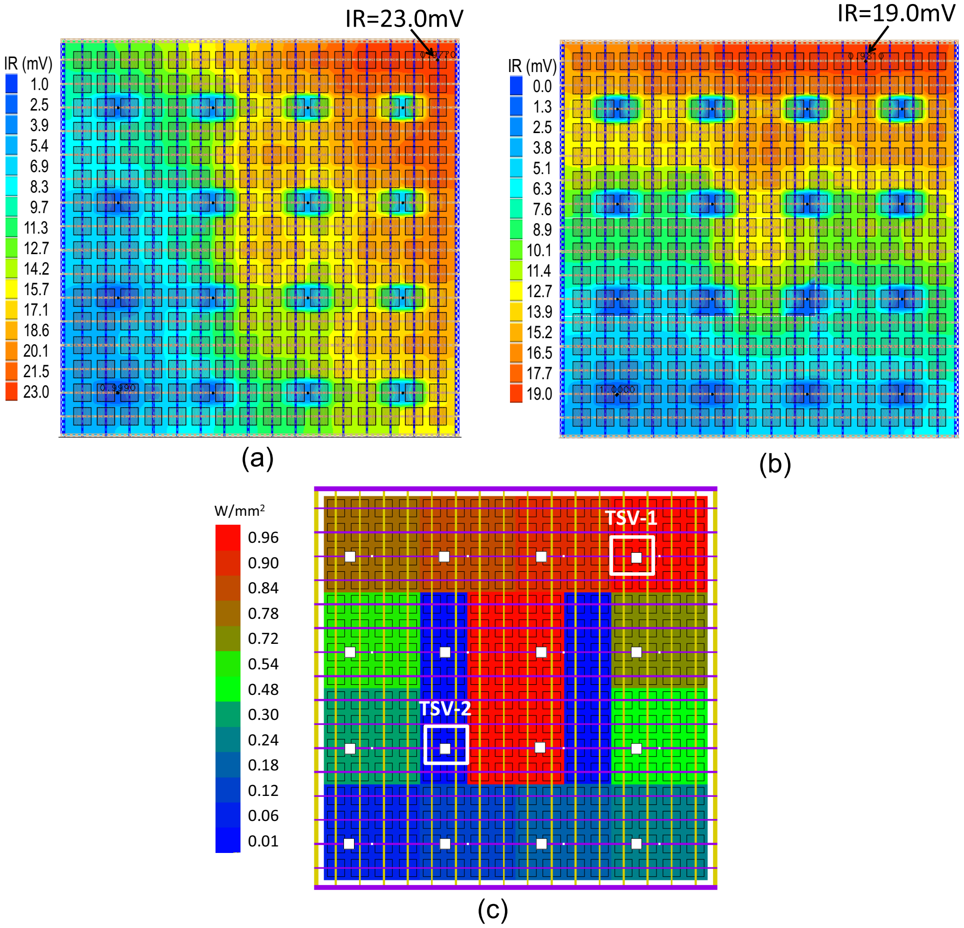


Figure 49: The voltage-drop maps in the top die (a) and in the bottom die (b). The power map in the bottom die (c).

TSVs: (1) The symmetric current density, e.g., TSV-1 in Figure 49(c), where the current density of all the power wires is high and (2) the asymmetric current density, e.g., TSV-2 in Figure 49(c), where the current density of the left power wires is much lower than the current density of the right power wires.

The voltage-drop maps in the top die and the bottom die are shown in Figures 49(a) and 49(b). The top-right corner has the maximum IR drop: 23.0mV IR drop in the top die and 19.0mV IR drop in the bottom. The IR drops in the bottom die are larger than the IR drops in the top die because of the TSV parasitic resistance. Since TSVs and C4s are aligned, the region close to TSVs has a smaller IR drop than the region far from TSVs.

Detailed current-density distributions in TSV-1 and TSV-2 are shown in Figure 50, where TSV-1, located in the center of the hot region, has fairly symmetric current densities along power wires; and TSV-2, located at the boundary between a power-hot region and a cool region, has asymmetric current densities in the power wires.

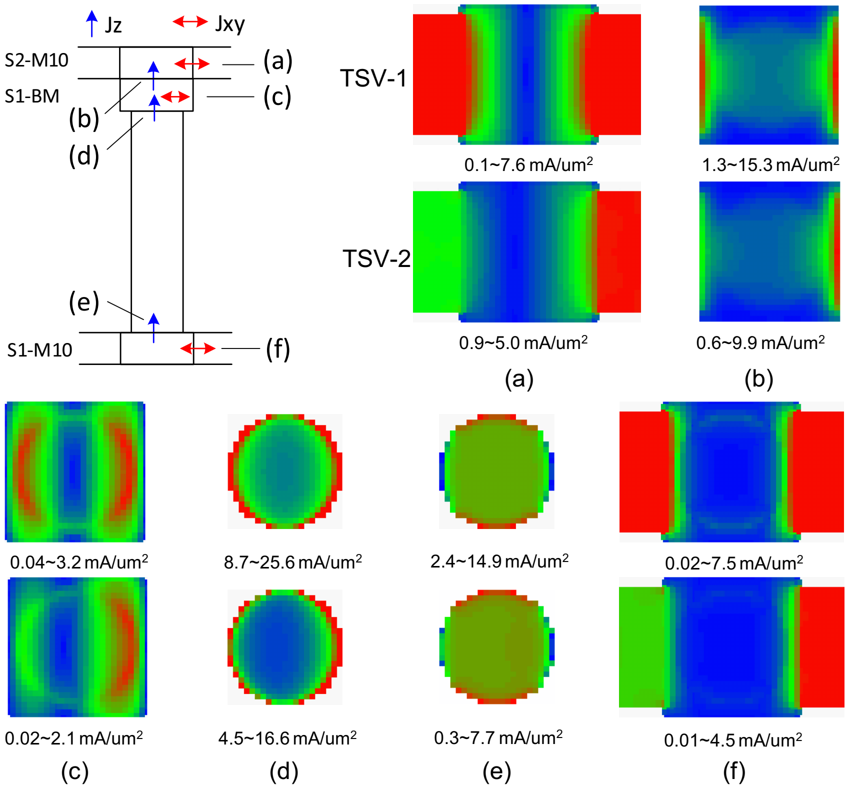


Figure 50: Current-density distribution in the XY direction (Jxy) and the Z direction (Jz) of TSV-1 and TSV-2.

The current densities in Metal 10 of Die-2 (S2-M10), back metal of Die-1 (S1-BM), and Metal 10 of Die-1 (S1-M10) are plotted in Figures 50(a), 50(c), and 50(f), respectively. The plots of Jz flowing through the interface between S1-BM and S2-M10, through the top surface of the TSV, and through the bottom surface of the TSV are depicted in Figures 50(b), 50(d), and 50(e), respectively.

First, PSIM is effective to capture the detailed current-density distribution inside the 3D power connections. A symmetric current crowding occurs at both edges of TSV-1, whereas most of the current crowds at the right edge of TSV-2.

Second, a large current crowding inside TSVs is observed. For TSV-1, the maximum current density (J_{max}) along the wire in Figure 50(a) is 7.6mA/μm^2, where most current concentrates at the connection between the power wire and the landing pad. However, the maximum current density through the TSV in the Z direction can reach to 25.6mA/μm^2 as shown in Figure 50(d), which is approximately 2.4 times larger than the wire J_{max}.

Third, a large current crowding occurs at the TSV top surface because of the aligned TSVs and C4s. The current density in the Z direction through the TSV bottom surface (Figure 50(e)) is 14.9mA/μm^2 compared with the top surface of 25.6mA/μm^2.

Fourth, in the bottom TSV surface (Figure 50(e)), the current in the Z direction crowds at the top and bottom edges instead of concentrating at the left and right edges. This phenomenon happens because a large amount of current in the XY direction flows out from the left and right edges to feed the current sinks in Die-1. As a result, the current, delivered to the power grid in Die-2, concentrates at the top and bottom edges. Moreover, the current crowding leads to a 5.7mV IR drop through TSV-1, which is 3.7% larger than the IR drop without considering the crowding.

The next subsections contain the following results: (1) The maximum current density (J_{max}) along the power wires, (2) the maximum and average current density (J_{avg}) of the TSVs, (3) the minimum, maximum, and average IR drops in top and bottom dies, and (4) the IR drop through the TSVs. A baseline PDN design contains a 16×16 power grid in each die, 16 TSVs, and 16 C4s. The TSV diameter is 5.0μm. The mesh size is 0.25μm.

6.3.3 Impact of TSV Mesh Size

To study the impact of the TSV mesh size on power integrity, the mesh size of the TSV model is increased from 0.25μm to 1.0μm. The results of the current density and the IR drop are shown in Table 15.

Table 15: Impact of the TSV mesh size on current density (mA/μm^2) and IR drop (mV). The TSV diameter is 5.0μm. And the power grid is 16×16.

#TSVs	Mesh	Wire	TSV w/ max(J_{max})			J_{inc} (%) of TSVs			IR_Bottom			IR_Top		
&#C4s	(μm)	J_{max}	J_{max}	J_{avg}	J_{inc}(%)	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
4×4	0.25	10.5	25.6	10.2	151	151	161	192	2.1	9.5	19.1	3.8	12.7	23.0
4×4	0.50	10.4	20.2	10.1	100	100	105	124	2.4	10.0	19.8	4.1	13.3	23.7
4×4	1.00	10.5	14.3	10.2	41	41	42	48	2.1	9.4	18.9	3.9	12.9	23.1

First, using large mesh tiles in the TSV model results in low J_{max}. As the mesh size increases from 0.25μm to 1.0μm, J_{max} reduces from 25.6mA/μm^2 to 14.3mA/μm^2, and J_{inc}

reduces from 151%-192% to 41%-48%. This phenomenon happens because the coarse mesh averages out the current gradient. Second, the mesh size does not affect the IR drop of power grids and the wire J_{max} very much. In contrast, a significant current crowding is observed in the TSVs for small mesh sizes. For the mesh size of 0.25µm, the TSV J_{max} is 25.6mA/µm^2, which is 110% larger than the TSV J_{avg} of 10.2mA/µm^2.

6.3.4 Impact of Power Wire Density

For the baseline PDN design and power maps described in Section 6.3.2, we increase the power grid from 8×8, 12×12, 16×16, to 20×20 and fix other design factors. The power wire density increases from 2.9% to 7.1% over the footprint area. The simulation results of current density and IR drop are shown in Table 16. Using more power wires helps to reduce the IR drop in both dies, but reduces the J_{max} of TSVs and wires in small scale. The maximum IR drop in bottom and top die reduces from 36.0mV to 15.2mV and 37.8mV to 20.6mV, respectively; the maximum J_{max} of the TSVs only reduces from 28.8mA/µm^2 to 25.2mA/µm^2. This is mainly due to the fixed placement of TSVs and C4s, where the current through each TSV in the Z direction is related to the TSV count.

Table 16: Impact of the power wire density on current density (mA/µm^2) and IR drop (mV). The TSV mesh size is 0.25µm, the TSV diameter is 5.0µm.

Power	P-wire	Wire	TSV w/ max(J_{max})			J_{inc} (%) of TSVs			IR_Bottom			IR_Top		
grid	den	J_{max}	J_{max}	J_{avg}	J_{inc}(%)	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
8x8	2.9%	11.8	28.8	11.2	157	147	169	223	3.1	15.7	36.0	4.8	18.5	37.8
12x12	4.3%	11.6	27.9	10.6	162	151	167	200	2.6	11.1	22.8	4.1	15.1	29.3
16x16	5.7%	10.5	25.6	10.2	151	151	161	192	2.1	9.5	19.1	3.8	12.7	23.0
20x20	7.1%	10.5	25.2	9.9	154	150	160	183	1.9	8.0	15.2	3.6	11.7	20.6

6.3.5 Impact of TSV and C4 Count

For the baseline PDN design and power maps described in Section 6.3.2, we increase the TSV and C4 count from 2×2, 3×3, 4×4, to 5×5 and fix other design factors. The simulation results of the current density and IR drop are shown in Table 17. Using more TSVs and C4s significantly reduces both the J_{max} and IR drop. With TSV and C4 count increases from 2×2 to 5×5, the J_{max} of wires reduces from 30.3mA/µm^2 to 8.4mA/µm^2; the J_{max} of the TSV reduces from 74.7mA/µm^2 to 19.8mA/µm^2; the worst IR in bottom and top die reduces from 59.5mV to 15.1mV, and 77.1mV to 18.3mV, respectively. This is mainly because using more TSVs leads to less current per TSV and thus lower J_{max}, and using more C4s helps to improve the IR. Furthermore, the J_{max} of the TSVs caused by current crowding is still around 150% to 190% larger over the J_{avg}, and approximately 140% larger over the J_{max} of the wires.

Table 17: Impact of the TSV count on current density (mA/μm^2) and IR drop (mV). The TSV diameter is 5.0μm, and the mesh size is 0.25μm

Power	#TSV	Wire	TSV w/ max(J_{max})			J_{inc} (%) of TSVs			IR_Bottom			IR_Top		
grid	&#C4	J_{max}	J_{max}	J_{avg}	J_{inc}(%)	min	avg	max	min	avg	max	min	avg	max
16x16	2x2	30.3	74.7	29.5	153	153	154	156	18.7	44.8	59.5	30.0	58.5	77.1
16x16	3x3	16.9	41.0	16.0	156	155	157	162	5.7	18.4	34.1	9.6	24.0	40.8
16x16	4x4	10.5	25.6	10.2	151	151	161	192	2.1	9.5	19.1	3.8	12.7	23.0
16x16	5x5	8.4	19.8	7.7	158	147	161	193	1.1	6.1	15.1	2.1	8.1	18.3

6.3.6 Impact of TSV Diameter

For the baseline PDN design and power maps described in Section 6.3.2, we increase the TSV diameter from 4μm, 5μm, 8μm, to 16μm, with mesh size of 0.25μm, 0.25μm, 0.5μm, and 0.5μm, respectively. Other design factors are fixed. The simulation results of current density and IR drop are shown in Table 18. We observe that larger TSVs significantly reduce J_{avg} from 15.7mA/μm^2 to 10mA/μm^2 and J_{max} of the TSVs from 33.5mA/μm^2 to 10.6mA/μm^2. However, J_{max} of TSVs reduces slower than J_{avg} of TSVs. As a result, for the 16μm diameter TSVs, J_{max} of TSVs is even 930% to 1180% larger than J_{avg} of the TSVs. In addition, the TSV diameter only affects IR drops in the top die. IR drops in the bottom die are insensitive to the TSV diameter because the voltage is directly supplied by C4s from the package. The top die has lower IR drops when using larger TSVs, which is due to the reduced TSV effective resistance and the IR through TSVs.

Table 18: Impact of the TSV diameter (μm) on current density (mA/μm^2) and IR drop (mV). The power grid is 4×4, and the mesh size is 0.25μm.

#TSV	TSV	Wire	TSV with max(J_{max})			J_{inc} (%) of TSVs			IR_Bottom			IR_Top		
&#C4	(μm)	J_{max}	J_{max}	J_{avg}	J_{inc}(%)	min	avg	max	min	avg	max	min	avg	max
4x4	4	10.5	33.5	15.7	113	109	113	117	2.2	9.5	19.2	4.3	13.6	24.1
4x4	5	10.5	25.6	10.2	151	151	161	192	2.1	9.5	19.1	3.8	12.7	23.0
4x4	8	10.4	19.0	4.0	372	372	394	463	2.3	9.9	19.7	3.3	11.8	21.7
4x4	16	10.7	10.6	1.0	928	928	986	1177	2.2	9.5	19.2	2.3	9.8	18.8

6.3.7 Impact of TSV and C4 Offset

Previous simulations assume aligned TSVs and C4s. To study the impact of offset on power integrity, a 175µm distance is created between the TSV and the C4. The offset design has 12 C4s and 16 TSVs. The simulation results are shown in Table 19.

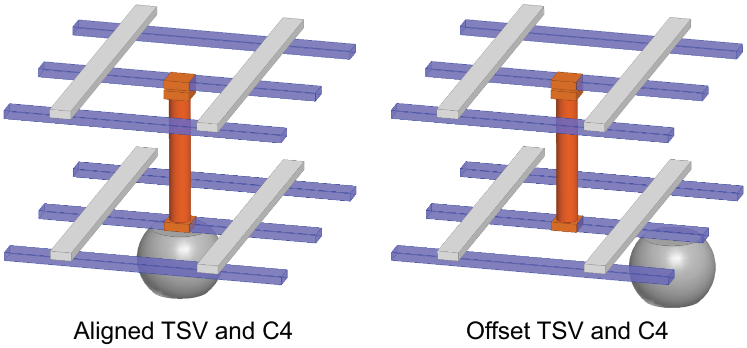


Figure 51: Zoom-in for partial PDNs with aligned vs offset TSV and C4.

The current crowding has larger impact on the TSV IR drop in the offset design than in the aligned design. In Table 19, the six columns from the right compare the IR drop through the TSV with (IR_c) and without (IR_n) considering current crowding. The current crowding in the offset design results in 5.9% to 10.6% larger IR drop than IR_n, whereas in the aligned design, current crowding results in 3.4% to 5.2% larger IR drop than IR_n. This phenomenon happens mainly because a large current crowding occurs in both the top and bottom surfaces of TSVs in the offset design, whereas in the aligned design, only the top interface between the TSV and the backside metal has a large current crowding, where the voltage at the bottom interface between TSVs and S1-M10 is constantly supplied by C4s.

6.3.8 3D Power Integrity on Large-Scale PDNs

Five large-scale two-die stacked PDNs are designed for 3D power analysis using PSIM. The power-wire utilization, defined as the total area of power wires in each die over the footprint area, is set to 5%. The local and global power density refers to the 3D core-to-memory PDN designs [75] [93], Intel microprocessors, and the power density estimation in the International Technology Roadmap for Semiconductors (ITRS) 2005 [6].

The results of power analysis on large-scale PDNs are shown in Table 20. First, excessive

Table 19: Impact of TSV and C4 offset on current density (mA/µm^2) and IR drop (mV) through TSVs.

		Aligned	Offset
#TSVs		16	16
#C4s		16	12
Wire J_{max}		10.5	25.2
TSV with max(J_{max})	J_{max}	25.6	22.0
	$J_{avg.}$	10.2	8.3
	$J_{inc.}$(%)	151	165
J_{inc}(%) of TSVs	Min	151	149
	Avg	161	164
	Max	192	190
IR_Bottom	Min	2.1	10.2
	Avg	9.5	26.5
	Max	19.1	48.3
IR_Top	Min	3.8	29.7
	Avg	12.7	46.7
	Max	23.0	65.5
TSV with max(IR)	IR_c	5.7	4.8
	IR_n	5.5	4.5
	Inc.(%)	3.7	6.4
Inc(%) of TSV IR	Min	3.4	5.9
	Avg	4.0	8.0
	Max	5.2	10.6

current densities through the TSVs are observed. The TSV J_{max} is 40% to 47% larger than the TSV J_{avg}. Second, the wire J_{max} is affected by the power density in Die-1 and Die-2. The large power density (PDN2 and PDN4) in the bottom die results in comparable J_{max} of wires and TSVs. When the power density in the bottom die is low, the TSV J_{max} is 13% larger than the wire J_{max}. Third, current crowding also increases the IR drop through the TSVs. The TSV IR drop with current crowding is 11.4% to 12.2% larger than the IR drop without considering current crowding. Furthermore, the IR drops in Die-1 and Die-2 are also affected by the power density. When each die has a comparable power density, the maximum IR in the top die is usually larger than that in the bottom die. Allocating high power densities close to C4s (in the bottom die) helps reduce the IR drops in the top die.

6.4 Summary

In this chapter, the current crowding inside TSV-based 3D power connections has been studied. First, the current-density distribution inside the 3D TSV-based power grids has been investigated. A large current gradient called current crowding near the interface

Table 20: Power integrity analysis for large-scale 3D PDNs including the footprint (mm^2), power density (W/mm^2), current density (mA/µm^2), and IR drop (mV).

Design		PDN1	PDN2	PDN3	PDN4	PDN5
Footprint		5×5	6×6	9×9	11×11	15×15
Power grid		50×50	60×60	90×90	110×110	150×150
#TSVs		144	225	484	729	1369
#C4s		144	225	484	729	1369
Power density	top	0.57	0.4	0.8	0.71	0.47
	bot	0.57	0.75	0.8	0.91	0.49
Wire J_{max}	top	7.0	3.5	13.6	8.7	16.2
	bot	7.2	6.6	12.1	11.4	17.4
TSV with max(J_{max})	J_{max}	9.6	5.0	18.5	11.1	23.3
	J_{avg}	6.8	3.6	13.1	7.5	16.3
	J_{inc}(%)	41	40	41	47	43
IR_Bottom	min	5.1	6.0	4.4	8.1	1.8
	avg	8.7	9.9	11.3	12.8	6.8
	max	15.9	13.3	24.2	25.2	34.9
IR_Top	min	7.9	5.0	6.8	9.7	2.7
	avg	11.7	7.2	15.6	13.5	8.8
	max	19.6	9.2	37.8	24.2	49.6
TSV with max(IR)	IR_c	4.1	2.1	7.9	5.1	9.9
	IR_n	3.7	1.9	7.1	4.5	8.8
	Inc.(%)	11.4	11.4	11.5	11.4	12.2

between power wires and TSVs has been observed. In addition, the current crowding also increases the effective resistance of the TSV and the voltage drop in the PDN. Second, a 3D TSV model has been implemented and simulated using PSIM. This model has a good accuracy and far less complexity compared with the finite-element tools. Third, PSIM with the proposed simple TSV model has been applied on chip-scale 3D PDNs to analyze detailed current-density distributions and voltage drops. By identifying the current crowding corner inside each TSV, PSIM helps assign reasonable current limits and voltage-drop limits for 3D PDN design and optimization. Moreover, PSIM can select a different mesh size depending on the resolution of the power analysis. First, for a large-scale PDN, a coarse mesh size can be used to quickly identify the hotspots associated with the maximum current density and IR drop. Then, in a bounded hotspot region, a fine mesh size can be used to identify the detailed current-density distribution and to optimize the power grid, correspondingly.

CHAPTER VII

MODELING OF ATOMIC CONCENTRATION AT THE WIRE-TO-TSV INTERFACE

Electromigration (EM) decreases the reliability of integrated circuits (ICs). It may eventually cause shorts or opens in circuits and interconnects which can reduce IC lifetimes, or worst, cause field fails. EM is driven by multiple physical mechanisms, including electric current, temperature gradient, stress gradient, and atomic concentration gradient. The evolution of atomic concentration or the mean time to failure (MTTF) are two important parameters to investigate the EM reliability. This analysis requires a transient analysis of the atomic concentration. Atomic diffusion is significantly different within a metal grain and along grain boundaries, each having different activation energies. Atomic transport is dominated by grain boundary diffusion and must be included in any realistic EM simulation.

Through-silicon-via (TSV)-based 3D integration has gained a lot of interest due to its potential to overcome conventional CMOS scaling limitations and its potential to enable heterogeneous integration. Reliability of TSV-based 3D ICs is an important issue for main stream acceptance. In particular, the reliability related to EM in 3D TSVs and TSV connections is a critical issue to explore. TSVs, especially the power/ground (P/G) TSVs in 3D power delivery networks (PDNs), carry large amounts of current. Specifically, P/G TSVs which typically have a high average current density can have much higher local current densities due to current crowding. These regions of high local current density are much more susceptible to EM degradation. Moreover, the large power density with high temperature or large thermal gradient inside 3D ICs due to multi-tier stacking or joule heating can accelerate atomic migration. Therefore, analyzing the evolution of atomic concentration and the EM lifetime for the 3D connection is important.

A test case to study the EM reliability of wire-to-TSV interface is shown in Figure 52.

A TSV having no grain structure is illustrated in Figure 52(a). Here the entire TSV is considered as a perfect crystal. However, in reality, most metal materials have polycrystalline structures with grains having a characteristic average size. These grains are separated by grain boundaries having a characteristic thickness. The TSVs shown in Figures 52(b) and 52(c) have simplified grains structures with sizes of 2.0um and 1.0um, respectively. The grain structure significantly affects the atom diffusion and the EM lifetime.

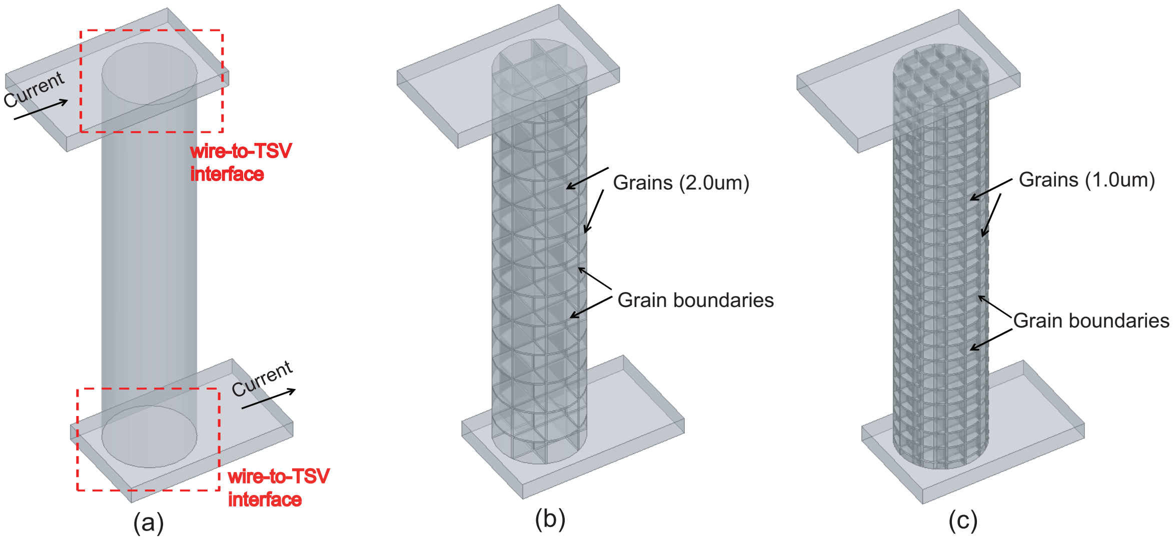


Figure 52: A test case to study the EM reliability of wire-to-TSV interface, with no grain structure (a), 2.0um grain size (b), and 1.0um grain size (c).

In this chapter, the atomic concentration in TSVs is modeled and analyzed. Investigations are performed on the impact of current crowding, grain structure, and temperature on EM lifetime using a multi-physics simulation. Transient analysis is applied on the atomic concentration and its evolution with grain and grain boundary structures. Current crowding at the wire-to-TSV interface accelerates the atomic migration and reduces the lifetime of TSVs. The impact of current, temperature, and grain structure on the EM lifetime of TSVs are explored. In addition, the TSV resistance change is modeled.

7.1 Fundamentals

7.1.1 Mean Time To Failure

The mean-time-to-failure (MTTF) is an important parameter used to characterize the time to potential failures during operation. Previous work utilized the following criteria to identify the MTTF subject to EM.

$$\text{MTTF} = Aj^{-n}e^{E_{\text{A}}/kT} \tag{26}$$

Equation (26), known as Black's equation [68] is the most commonly used method to predict the life span of integrated circuits due to EM. It enables accelerated EM testing, where the coefficient A, scaling factor n, and the activation energy E_{A} are determined by fitting the model to the experimental data, and k is the Boltzmann's constant. This equation clearly shows that the EM failure rate depends exponentially on the temperature T and depends on a power of the current density j. However, this model does not include the thermal migration caused by thermal gradients, and is not based on a specific physical model. Thus, it is hard to identify the potential failure locations.

Another method to locate the EM sensitive regions is to calculate the atomic flux divergence (AFD) [15, 94] at each location using the finite element model (FEM) approach. The maximum AFD is usually considered as a likely failure site. The atomic fluxes are calculated using Equations (29) through (32), where the initial atomic concentration is N_0. However, the maximum AFD is a stationary analysis result, which can not predict the atomic concentration and its evolution over time.

Our modeling approach is based on the FEM approach. The atomic concentration is solved by using the partial differential equations, Equations (27) through (32). Detailed discussions will be presented in Section 7.2. In this paper, we report the MTTF when the atomic concentration has 10% deviation of the initial value.

7.1.2 Grains and Grain Boundaries

The grain structure depends on the pretreatment of copper and on the conditions of deposition. This grain structure of the conductive material has a strong influence on the lifetime. Theoretically, the crystal structure of a solid material, e.g., copper, should have a perfect periodic arrangement of "face centered cubic" structure. However, in reality, most metals never consist of only one crystal, but contain a collection of small crystals, so called polycrystalline structure as shown in Figure 53. Each small crystal, called a grain, has periodic arrangement of atoms. The average diameter of grains is called grain size. Inside each grain, momentum exchange between the electrons and atoms is small because of the uniform lattice structure of metal ions. However, the periodic pattern is broken at the interface between two grains, called a grain boundary. The atoms in this transition region can not match up perfectly with both crystal lattice, so the momentum transfer in the grain boundary is much larger.

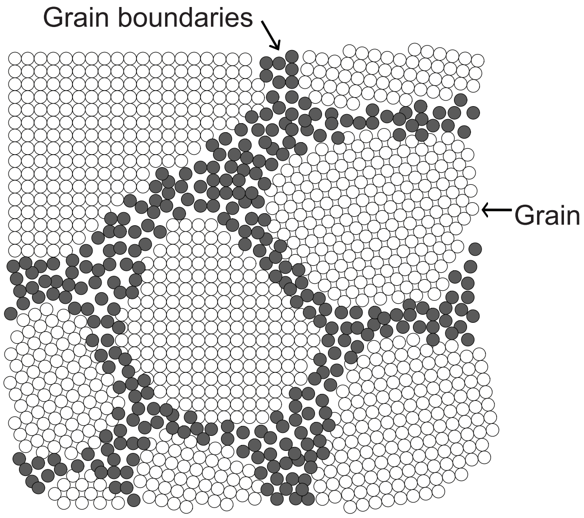


Figure 53: Illustrations of grains and grain boundaries in polycrystalline.

Since the atoms are bounded weakly in the grain boundaries, once a strong force is applied, such as concentration gradient, thermal gradient, current, or stress gradient, the atoms become mobile. The diffusion caused by EM includes lattice diffusion, surface diffusion, and grain boundary diffusion. Since the diffusion barrier layer between the TSV and the silicon dioxide typically helps minimize the migration of TSV metal into the silicon, our model mainly focuses on the lattice diffusion and grain boundary diffusion of the TSVs. Correspondingly, the lattice has high activation energy E_{A}, whereas the grain boundaries have low E_{A}.

7.2 Modeling Approach and Settings

A set of partial differential equations (PDE) is used to obtain the atomic concentration, $N(x, y, z, t)$, at each location (x, y, z) at time t, where the atomic concentration evolution is described in a continuity equation (Equation (27)), and the atomic flux $\mathbf{J}$ is determined by the combined mechanisms of concentration gradient ($\mathbf{J_N}$ in Equation (29)), current density ($\mathbf{J_c}$ in Equation (30)), thermal gradient ($\mathbf{J_T}$ in Equation (31)), and stress gradient ($\mathbf{J_s}$ in Equation (32)).

These PDEs are formulated and solved in COMSOL multiphysics simulation tool [95]. In addition, the current density, temperature, and hydrostatic stress can be also obtained by electrical-thermal and thermal-mechanical coupling simulations in COMSOL.

7.2.1 Electromigration Equations

The PDEs used to obtain the atomic concentration are shown as follows.

$$\frac{\partial N}{\partial t} + \nabla \cdot \mathbf{J} = 0 \tag{27}$$

$$\mathbf{J} = \mathbf{J_c} + \mathbf{J_T} + \mathbf{J_s} + \mathbf{J_N} \tag{28}$$

$$\mathbf{J_N} = -D \ \nabla N \tag{29}$$

$$\mathbf{J_c} = \frac{N}{kT} eZ^* \rho \mathbf{j} \ D \tag{30}$$

$$\mathbf{J_T} = -\frac{NQ^*}{kT^2} \ D \ \nabla T \tag{31}$$

$$\mathbf{J_s} = \frac{N\Omega}{kT} \ D \ \nabla \sigma_{\mathrm{H}} \tag{32}$$

$$D = D_0 \ \exp(\frac{-E_{\mathrm{A}}}{kT}) \tag{33}$$

where N is the atomic concentration per unit volume, which is the variable in the PDEs. N_0 is the initial concentration. ∇N is the concentration gradient. The atomic diffusion D is described as $D_0 \ \exp(\frac{-E_{\mathrm{A}}}{kT})$, where D_0 is the self-diffusion coefficient, k is the Boltzmann constant, T is the absolute temperature. The diffusion $\mathbf{J}$ is the total atomic flux at a location, which includes the flux caused by concentration gradient $\mathbf{J_N}$, current density $\mathbf{J_N}$, thermal gradient $\mathbf{J_T}$, and stress gradient $\mathbf{J_s}$. $\mathbf{j}$ is the current density. ∇T is temperature gradient. $\nabla \sigma_{\mathrm{H}}$ is the hydrostatic stress gradient. The meanings of other notations are summarized in Table 21. Detailed discussions of these PDEs will be presented in the following subsections of this paper.

Table 21: Notations and meanings in EM PDEs.

term	meaning
N	Atomic concentration in atoms/m^3
$\mathbf{j}$	Current density in mA/um^2
E_{A}	Activation energy in eV
k	Boltzmann constant in J/K
T	Absolute temperature in K
e	Electric charge in C
Z^*	Effective valence charge
ρ	Electrical resistivity in Ω·m
D_0	Self-diffusion coefficient in m^2/s
Q^*	Heat of transport
Ω	Atomic volume in m^3
σ_{H}	Hydrostatic stress in Pa

7.2.2 Atomic Flux and Atomic Flux Divergence

Atomic flux, $\mathbf{J}$, describes the total number of atoms that flow across a unit area per unit time. A large atomic flux means the atoms moves fast across the unit area. Atomic flux divergence, $\triangledown \cdot \mathbf{J}$, describes the changes of atomic number through unit volume per unit time, which is the spatial difference between the inward and outward flux at the boundary planes of the unit volume.

Equation (27) is the continuity equation that describes the atomic concentration evolution over time and also insures that atoms are conserved. It governs the atomic flux divergence over spatial dimensions and determines how the atomic concentration evolves over time. As shown in Figure 54, along the direction x, when the inward atomic flux $\mathbf{J_{in}}$ is larger than the outward flux $\mathbf{J_{out}}$, the atoms in the unit volume tends to increase. From Equation (27), the atomic flux decreases over the x direction, which corresponds to a negative atomic flux divergence. As a result, the $\frac{\partial N}{\partial t}$ will be positive for mass conservative. This means the atomic concentration N in this unit volume tends to increase over time.

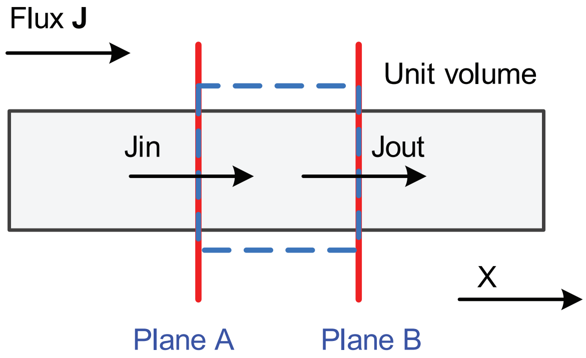


Figure 54: Illustration of the atomic flux and divergence.

7.2.3 Effect of Activation Energy and Atomic Concentration

If we set $\mathbf{J_c}$, $\mathbf{J_T}$, $\mathbf{J_s}$ to zero, atomic diffusion can still occur due to the atomic flux $\mathbf{J_N}$ from the atomic concentration gradient and the difference of activation energy in grains and grain boundaries. Equation (29) is analogous to the Fick's first law. The flux is proportional to the negative concentration gradient $-\nabla N$. The atoms are likely to flow from high-concentration regions to low-concentration regions.

The activation energy E_{A} is different in grains $E_{\mathrm{A}}(\mathrm{g})$ and grain boundaries $E_{\mathrm{A}}(\mathrm{gb})$. Smaller $E_{\mathrm{A}}(\mathrm{gb})$ in grain boundaries results in high diffusion, whereas larger $E_{\mathrm{A}}(\mathrm{g})$ in grains results in low diffusion. Because D exponentially depends on $-E_{\mathrm{A}}$ (shown in Equation (33)), a large divergence of diffusion in grains and grain boundaries can be observed, which leads to large atomic flux divergence. Therefore, atomic accumulation or depletion may be observed around the grain boundaries.

7.2.4 Effect of Current

Atomic flux caused by electric current density is governed by Equation (30), where e is the electron charge, ρ is the resistivity of the conductor, $\mathbf{j}$ is the local current density.

In the present of non-zero local current density, thermally activated metal ions are acted on by two opposing forces as shown in Figure 55, which can be described as the effective valence Z^*. The Z^* consists of two parts, $Z^* = Z^*_{el} + Z^*_{wd}$, where $Z^*_{el}eE$ is the direct electrostatic force on the positive ions as a result of the electric field E. This force has the same direction as electric filed, but opposite to the electron flow. The $Z^*_{wd}eE$

is called the electron wind force, which is caused by the momentum exchange between conducting electrons colliding with the activated metal ions, and is in the opposite direction as $Z_{el}^{*}eE$. The diffusion of the atom is found to be enhanced in the direction of electron wind, so the momentum exchange effect is much greater than the electrostatic filed effect for electromigration in metals [96].

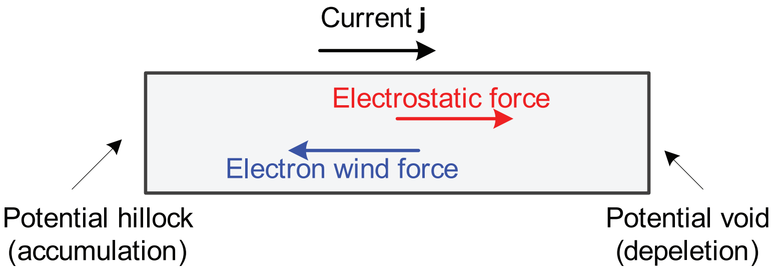


Figure 55: The electrostatic force and electron wind force on the atoms, and the weak positions of void and hillock formation.

7.2.5 Effect of Thermal and Stress

Atomic flux caused by thermal gradient is shown in Equation (31), where Q^* is the heat of transport. This means that the atomic flux is proportional to the negative gradient of temperature ($-\nabla T$), and atoms likely to move from high-temperature regions to low-temperature regions.

In addition to the effect of thermal gradient, the temperature profile also exponentially impacts the atom diffusion D in each atomic flux $\mathbf{J_c}$, $\mathbf{J_N}$, and $\mathbf{J_s}$. That is, the high temperature will accelerates the atomic diffusion, thus shortening the lifetime.

Atomic flux caused by hydrostatic stress gradient is shown in Equation (32), where Ω is the atomic volume, $\sigma_{\mathrm{H}} = (\sigma_x + \sigma_y + \sigma_z)/3$, σ_x, σ_y, σ_z are the corresponding normal stresses in the Cartesian coordinates system (x, y, z). The $\nabla\sigma_{\mathrm{H}}$ is the stress gradient that results from material accumulation and depletion due to electromigration. The stress involves both EM-induced back-flow mechanical stress [97] and the residual stress that is generated in thermal processing when there is a difference in thermal expansion coefficients (CTE) in the TSV structure [14]. We found $\mathbf{J_s}$ which related to CTE mismatch is small compared with the other atomic flux ($\mathbf{J_T}$, $\mathbf{J_c}$, $\mathbf{J_N}$) components and will therefore ignore $\mathbf{J_s}$ in the simulations in the rest of this paper.

7.2.6 Model Settings

The structure to investigate the atomic concentration evolution and EM lifetime is shown in Figure 52, which consists of the following components: (1) a copper TSV with 5um diameter and 25um height, (2) landing pads are 6um×10um and 1um thick, (3) a TSV is composed of regular cubic grains and grain boundaries. A current sources is inserted at the top-left corner; the current sink is defined at the bottom-right corner. This test case constrains the current flow direction and helps us investigate the current density impact on atomic concentration and EM reliability with the presence of grain boundaries.

COMSOL multiphysics is used to simulate the DC current density distribution, temperature distribution, and stress distribution, and to solve the partial differential equations to obtain the atomic concentration over time.

Our special interest is to investigate the atomic concentration and EM reliability at the wire-to-TSV interface, where large current crowding happens. The MTTF is defined as the 10% deviation of atomic concentration over the initial concentration inside the TSV, where the initial concentration N_0 is 1.53e28 Atoms/um^3. Since we focus at the TSV-to-wire interface, the wire is assumed a perfect diffusion model, where no grain is modeled inside the wire. In reality, if the depletion or accumulation of atomic concentration is expected at a specific TSV-wire interface, then both the TSV and the wire are expected to have voids or hillocks. Meanwhile, the grain structure of the wire will also affect the atomic concentration.

The TSV structure contains regular cubic grains and grain boundaries. Since no measurement data has been reported on the TSV grain structure, which can be different from Cu interconnects, we vary the grain size from 2.0um to 1.0um to study its impact on MTTF. The default grain size is 0.9um with a grain boundary thickness of 0.1um. The activation energy of the lattice (grain) is 2.1eV; and the grain boundary has default activation energy as 0.8eV, which will vary from 0.7eV to 0.9eV [98] to investigate its impact on MTTF.

Unless specified, other default values in the models are as follows: the input current density inside the TSV is 3.1mA/um^2; the temperature is 350K; the Z^* is -4; k is 1.38e-23 J/K, e is 1.6e-19C, $\rho = \rho_0(1 + \alpha(T - T_0))$, where ρ_0 is 1.68e-8Ω·m, α is 0.0039, T_0 is 293K. the D_0 is 1e-7 m^2/s, Q^* is 1.387e-20, Ω is 1.182e-29m^3.

7.3 Simulation Flow and Assumptions

7.3.1 Simulation Flow

In this study, we use a commercial tool, COMSOL MULTIPHYSICS, to conduct the simulation, which is superior to customize and solve the partial differential equations. The simulation flow is illustrated in Figure 56.

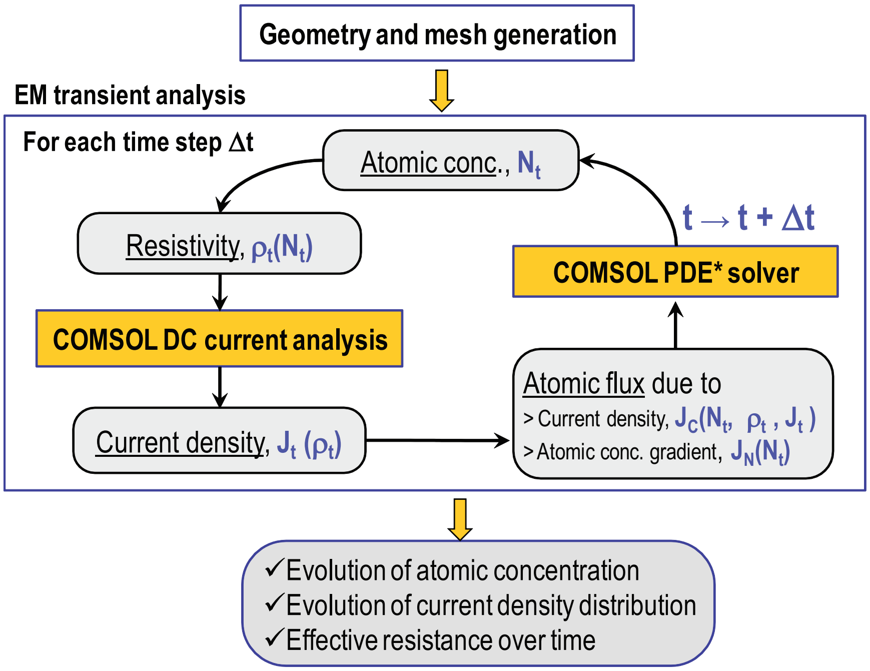

Geometry and mesh generation
EM transient analysis
For each time step Δt
Atomic conc., N$_t$
Resistivity, ρ$_t$(N$_t$)
COMSOL DC current analysis
Current density, J$_t$ (ρ$_t$)
t → t + Δt
COMSOL PDE* solver
Atomic flux due to
> Current density, J$_C$(N$_t$, ρ$_t$, J$_t$)
> Atomic conc. gradient, J$_N$(N$_t$)
✓Evolution of atomic concentration
✓Evolution of current density distribution
✓Effective resistance over time

Figure 56: Simulation flow using COMSOL.

This flow starts from creating geometry and generating meshes. Then the EM transient analysis is performed, which consists of an iteration loop. At each time step Δt, the atomic concentration at current time t, $\mathrm{N_t}$, is given. The resistivity distribution $\rho_{\mathrm{t}}(\mathrm{N_t})$ is calculated based on the resistivity function, which describes the resistivity based on the local atomic concentration. After COMSOL DC current analysis, the current density distribution $\mathrm{J_t}(\rho_{\mathrm{t}})$ can be simulated, which is a function of resistivity. Then the atomic fluxes can be updated, where the flux driven by the current density $\mathrm{J_c}$ is a function of the current atomic concentration $\mathrm{N_t}$, the resistivity distribution ρ_t, and the current density distribution $\mathrm{J_t}$. The flux driven by atomic concentration gradient $\mathrm{J_N}$ depends on the atomic concentration $\mathrm{N_t}$. The PDEs are then solved by COMOSL to obtain the atomic concentration in the next time $\mathrm{t + \Delta t}$. In this transient analysis, COMOSL determines the iteration times and step for convergence automatically.

7.3.2 Assumptions in This Work

A lot of assumptions are included in this modeling work. In principle, most of these assumption can be included in our model later. We assume uniform grain and grain boundary geometry. This model did not consider the grain orientation, grain/grain boundary propagation. No nucleation sites for void and hillock formation. The activation energy is obtained from the literature on wire structure because few works reported the activation energy of TSVs. We assume that grain and grain boundaries have the same initial atomic concentration and resistance. Our simulation shows negligible thermal gradient from joule heating and thermal stress because the copper TSV has very good thermal conductivity. Thus, we assume uniform temperature in the 3D structure. Diffusion is assumed stress independent. No quantum effect is considered, no atomic tunneling through grain boundaries. We didn't include back flow stress. Resistivity function is assumed dependent on atomic concentration. both grain and grain boundary is assumed the same function of resistivity.

7.4 Investigations on TSVs

7.4.1 Impact of Current Crowding

A recent work [70] analyzed the current density distribution for the 3D connection of wires and P/G TSVs. They discovered that for some geometries significant current crowding can occur giving rise to high local current densities at the wire-to-TSV interface. To analyze the impact of current crowding on atomic concentration, in this paper, we assume 60mA current flows from the top-left landing pad, through a TSV, and flows out of the bottom-right landing pad. The atomic concentration is affected by both atomic flux from current density ($\mathbf{J_c}$ in Equation (30)) and from atomic concentration gradient ($\mathbf{J_N}$ in Equation (29)).

Therefore, we include these two terms of flux in the continuity equation (27) and set $\mathbf{J_s}$ and $\mathbf{J_T}$ to zero.

The atomic concentration on the top and bottom wire-to-TSV interfaces at time 1e5s, 1e7s, and 1e8s are shown in Figure 57, where the color legend displays the percentage difference of the atomic concentration compared with the initial concentration (N_0=1.53e28 Atoms/m^3).

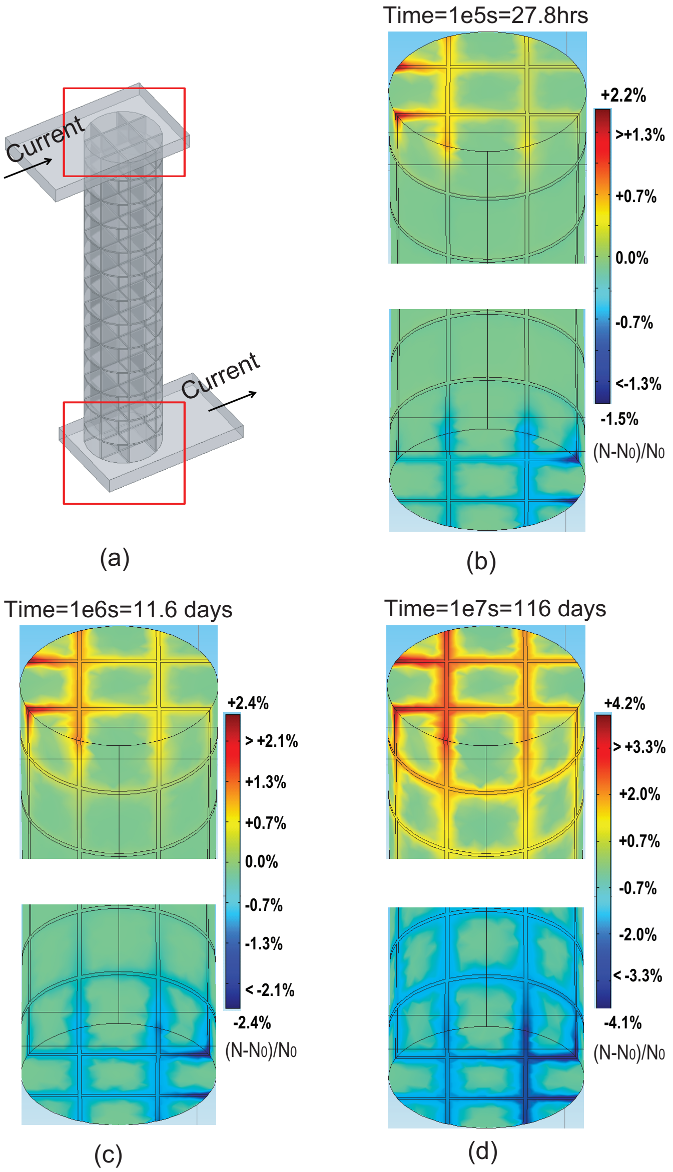


Figure 57: Atomic concentration on top and bottom wire-to-TSV interface at time=1e5s (b), time=1e6s (c), and time=1e7s (c). The color legend displays the percentage difference of atomic concentration normalized to the initial concentration (N_0=1.53e28 Atoms/m^3).

First, we observe that the atoms begin to accumulate/deplete along the grain boundary at time=1e5s. This accumulation or depletion penetrate to the neighboring grains over a short distance. This is because that grain boundaries provide fast paths with low activation energy for atom diffusion. Second, we observe that most accumulation (red) occurs at the top-left interface; and most depletion (blue) happens at the bottom-right interface. For example, at time=1e5s, the maximum atomic concentration is 2.2% larger than the initial value, whereas the minimum atomic concentration is 1.5% smaller than the initial concentration. In addition, these accumulation and depletion densities grow over time, and will very likely cause hillocks and voids, respectively. From time 1e5s to 1e8s, the maximum deviation of the atomic concentration compared with initial value increases from 2.2% to 4.2%. That is mainly due to the current crowding at each location and fast diffusion along grain boundaries. The local high current density increases the atomic flux and enlarges the atomic flux divergence as indicated in Equation (30).

Meanwhile, the current crowding at the wire-to-TSV interface is significantly affected by the thickness of landing wires [70]. A thinner landing wire causes larger current crowding than a thicker landing wire at the corners of wire-to-TSV interface. As a result, the maximum current density increases at these interfaces.

To investigate the current crowding impact on atomic concentration, we vary the landing wire thickness from 0.5um, 1.0um, 1.5um, 2.0um, to 3.0um. Meanwhile, the TSV diameter is kept at 5.0um, and the total current is 60mA. The impact of wire thickness on current density distribution and atomic concentration at time 1e7s for 0.5um and 3.0um thick wires are also shown in Figure 58.

Figures 58(a) through 58(c) are the 3D structure, current density distribution in side view and top/bottom wire-to-TSV interfaces, and atomic concentration in side view and top-bottom wire-to-TSV interfaces for 0.5um thick wire, respectively. Figures 58(d) through 58(f) are those for 3.0um wire thickness. The color legend of atomic concentration is the percentage difference of the concentration over the initial value.

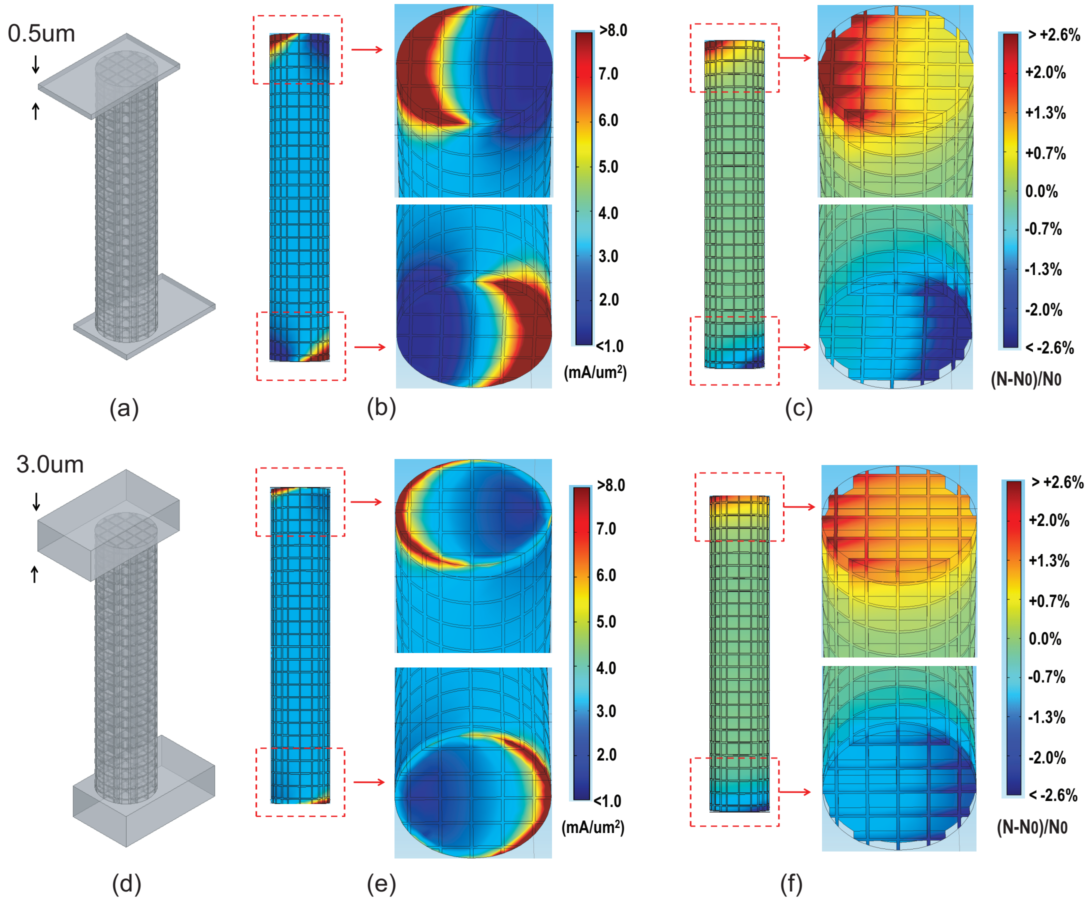


Figure 58: Impact of wire thickness on current crowding and atomic concentration at time 1e7s for top and bottom wire-to-TSV interfaces. The wire thickness is 0.5um (a)-(c) and 3.0um (d)-(f). (a) and (d) are 3D views for 0.5um and 3.0um wire thickness. (b) and (e) are current density distributions in side view and in 3D top and bottom wire-to-TSV interfaces for 0.5um and 3.0um wire thickness. (c) and (f) are atomic concentrations in side view and in 3D top and bottom wire-to-TSV interfaces for 0.5um and 3.0um wire thickness. The color legend of atomic concentration is the percentage difference normalized to the initial concentration N_0=1.53e28 atoms/m^3.

In Figures 58(b) and 58(b)(e), we observe significant current crowding at both top and bottom corners of wire-to-TSV interfaces. Thinner wires result in more current crowding. These results are consistent with the current crowding discussed in other papers [70]. The atomic concentration distributions, shown in Figures 58(c) and 58(f), demonstrate that more atoms accumulate at the top-left and deplete at the bottom-right, where current crowding gives higher current densities. This implies that using thin wires may result in earlier EM failures than using thick wires. In addition, in the case of 3.0um thick wire, since less current crowding occurs at the corners, the atom accumulation and depletion are spread over the entire interface but with lower local density.

Detailed results of maximum current density (Jmax) and average current density (Javg) inside the TSV, the atomic concentration at time t=1e7s, and MTTF are shown in Table 22. As the wire thickness decreases from 3.0um to 0.5um, the maximum current density inside the TSV increases from 11.0mA/um^2 to 37.1mA/um^2, however, the average current density remains at 3.1mA/um^2. Meanwhile, the maximum atomic concentration increases from 1.57e28 Atoms/m^3 to 1.63e28 Atoms/m^3, which corresponds to 2.6% to 6.5% larger concentration than the initial value; the minimum atomic concentration decreases from 1.49e28 Atoms/m^3 to 1.44e28 Atoms/m^3, which corresponds to 2.6% to 5.9% smaller concentration than the initial one; and the MTTF decrease from 3.0e8s to 0.3e8s. Note that the total input current is kept constant for each case. Current crowding can have a large impact on atomic concentration generating voids and hillocks and therefore accelerate EM failure.

Table 22: Impact of wire thickness on current density inside the TSV (mA/um^2), atomic concentration (Atoms/m^3) at time=1e7(s), and MTTF (s). Initial concentration is 1.53×10^{28} Atoms/m^3.

Wire	Current density		Atomic conc.($\times 10^{28}$)		MTTF
thickness (um)	Jmax	Javg	Max	Min	($\times 10^8$)
0.5	37.1	3.1	1.63	1.44	0.3
1.0	32.0	3.1	1.60	1.46	1.6
1.5	22.6	3.1	1.59	1.47	2.1
2.0	13.5	3.1	1.58	1.48	2.5
3.0	11.0	3.1	1.57	1.49	3.0

7.4.2 Impact of Current Direction and Density

The current direction determines the location of voids and hillocks. From Figure 57, we observe that the bottom-right wire-to-TSV interface has smaller concentration than the initial value (atom depletion) and the top-left wire-to-TSV interface has larger concentration than the initial value (atom accumulation). This means atoms move from the bottom-right corner to the top-left corner, which is opposite to the direction of positive current. This makes sense from a physical point of view and is due to the momentum exchange from the electrons to the atoms, which is the dominant force in EM, the atoms are pushed in the same direction as electrons (i.e., the opposite direction of current). Over time, atoms accumulate forming voids where current is injected. Likewise, atoms deplete forming voids where current is removed.

As the average current density inside the TSV is increased from 1.5mA/um^2 to 6mA/um^2 with the temperature set to 350K, the resulting MTTF is shown in Figure 59. The EM lifetime of a TSV with high current dramatically reduces from 2.6e9s to 1.0e7s. A TSV, carrying a high current density accelerates the depletion and accumulation of atoms, and decreases the EM lifetime. For P/G TSVs, which can carry the current density larger than 5mA/um^2, the EM reliability may become critical.

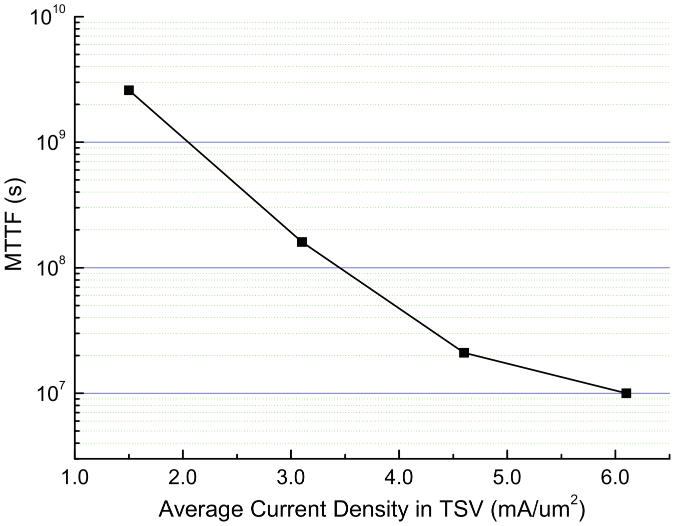


Figure 59: MTTF vs. average current density. The average current density increases from 1.5mA/um^2 to 6mA/um^2, T=350K.

7.4.3 Impact of Temperature

Temperature also plays an important role in atomic concentration and EM reliability. From Equations (28) to (29), the diffusivity D is exponentially related to the temperature. Moreover, Equation (31) also shows that atomic flux is affected by the thermal gradient. Note that, in 3D operation, the temperature can vary from tens of degree C to a hundred of degree C.

Joule heating from high current density inside a TSV causes high temperature. However, due to high thermal conductivity of copper, the thermal gradient is very small inside the TSV. The thermal gradient caused by joule heating of the TSV with 60mA input current is shown in Figure 60. The structure consists of three silicon layers (each is 25um thick), two inter-layer dielectric (ILD) layers (each is 4um thick), a TSV liner (SiO_2 with 0.2um thick), and a copper TSV with two landing wires. The heat sink is assigned at the top surface with the heat transfer coefficient of 25e3W/(m^2·K).

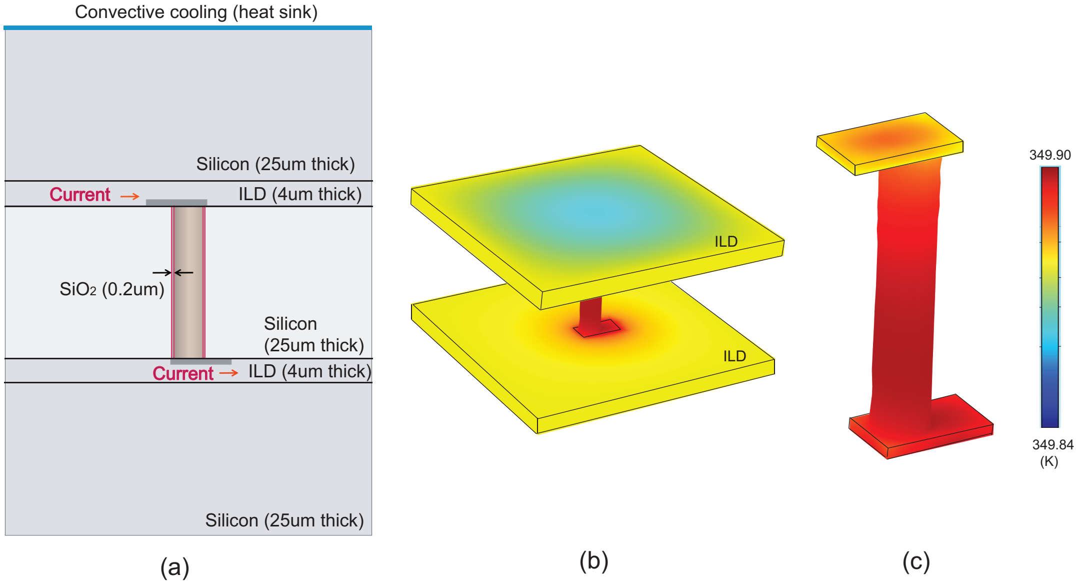


Figure 60: Simulation of joule heating for a TSV with 60mA input current. The structure (a) consists of three silicon layers, two ILD layers, a TSV liner (SiO_2), and a TSV with two landing wires. Heat sink is assigned at the top surface. (b) is the thermal gradient in ILD layers, landing wires, and the TSV. (c) is the thermal gradient inside the TSV which is negligible with a small range of 349.90K to 349.86K.

A small thermal gradient is shown in Figure 60(b) in the ILD layers, landing wires, and the TSV, where the temperature varies from 349.84K to 349.90K. The thermal gradient inside the TSV and landing wires is shown in Figure 60(c), which covers a small range of 349.86K to 349.90K. Therefore, we include the flux caused by current $\mathbf{J_c}$ and concentration gradient $\mathbf{J_N}$ in continuity Equation (27), and set other two terms of temperature gradient $\mathbf{J_T}$ and stress gradient $\mathbf{J_s}$ to zero.

To analyze the impact of temperature on migration, the current value is kept constant, and the temperature is increased from 300K to 400K. This temperature range is affected by both the power density from neighboring devices and the joule heating of the TSV. The impact of temperature on EM lifetime is shown in Figure 61. As the temperature increases from 300K to 400K, the MTTF is dramatically reduced from 5.9e9s to 8.7e6s.

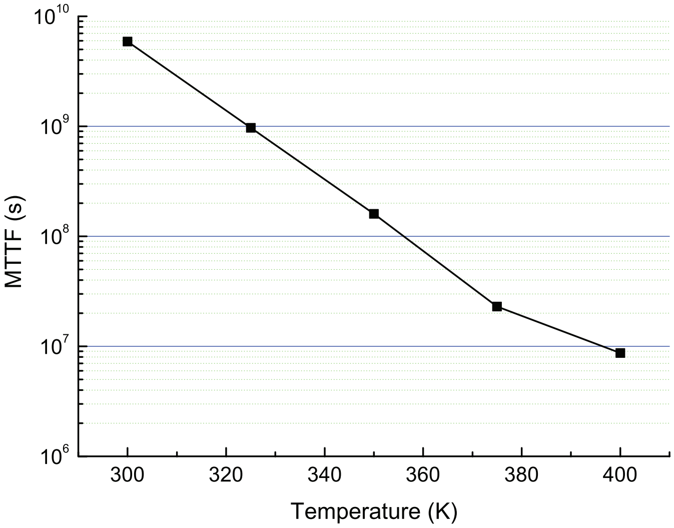


Figure 61: MTTF vs. temperature. The temperature is varied from 300K to 400K, and the current density is 3.1mA/um^2.

7.4.4 Impact of Grain Size

The grain structure and size is mainly determined by the manufacturing process, and can vary over a wide range. To study this, we vary the grain size of the TSV from 1.9um to 0.9um, while the grain boundary thickness is kept at 0.1um. The total current is 60mA, and the temperature is 350K.

The resulting MTTF is shown in Figure 62. With the grain size increases, the MTTF is increased from 1.6e8s to 3.1e8s. A TSV with larger grains helps to increase the lifetime. This is because the total grain boundaries with fast diffusion path decreases. Of course the average grain sizes and average grain boundary thicknesses can vary more than we have shown in this simple simulation. However, all these details can all be added to the model as needed.

7.4.5 Impact of Activation Energy

For these simulations, the activation energy of the grains and grain boundaries may also vary a lot. Especially, the small activation energy of grain boundaries determines the EM lifetime. Therefore, we increase the activation energy of grain boundaries from 0.7eV to 0.9eV to investigate its impact. The resulting MTTF is shown in Figure 63. We observe that with the activation energy reduces from 0.9eV to 0.7eV, the MTTF dramatically reduces

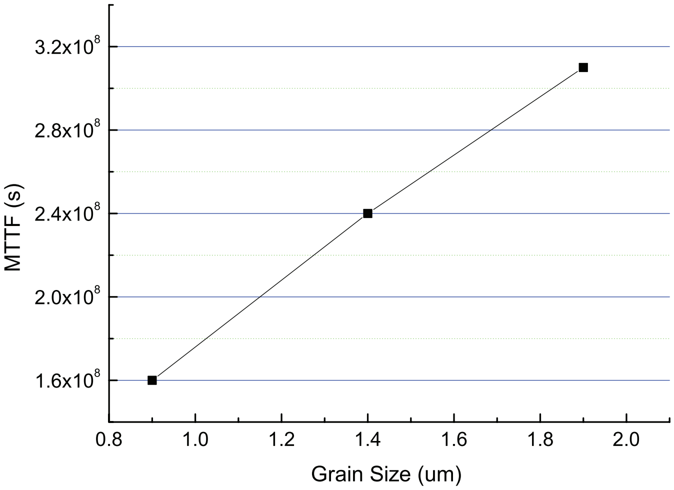


Figure 62: MTTF vs. grain size.

from 3.55e9s to 5.2e6s. This demonstrates the exponential impact of E_A on the atomic flux.

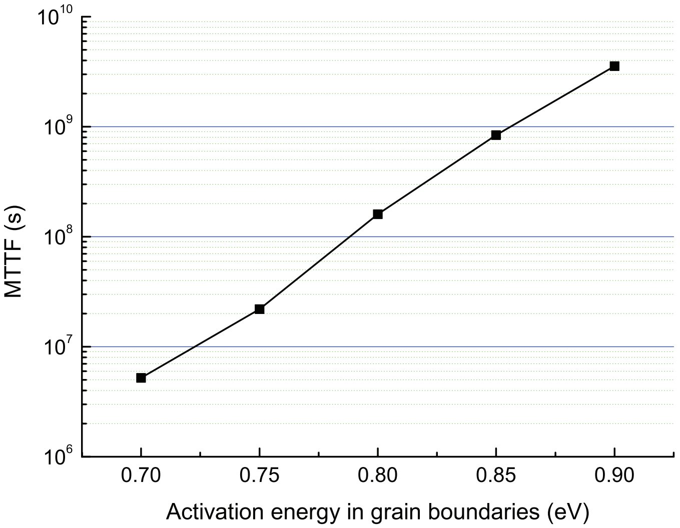


Figure 63: MTTF vs. activation energy in grain boundaries. Grain size and grain boundary size is 0.9um and 0.1um.

7.5 Simulation of TSV Effective Resistance

7.5.1 Resistivity Function

To simulate the effective resistance of the TSV-based 3D connection, we need to construct a resistivity function. Since we could not find the resistivity function in the literature, we arbitrarily defined a function to describe the resistivity evolution w.r.t. the atomic concentration, which is depicted in Figure 64.

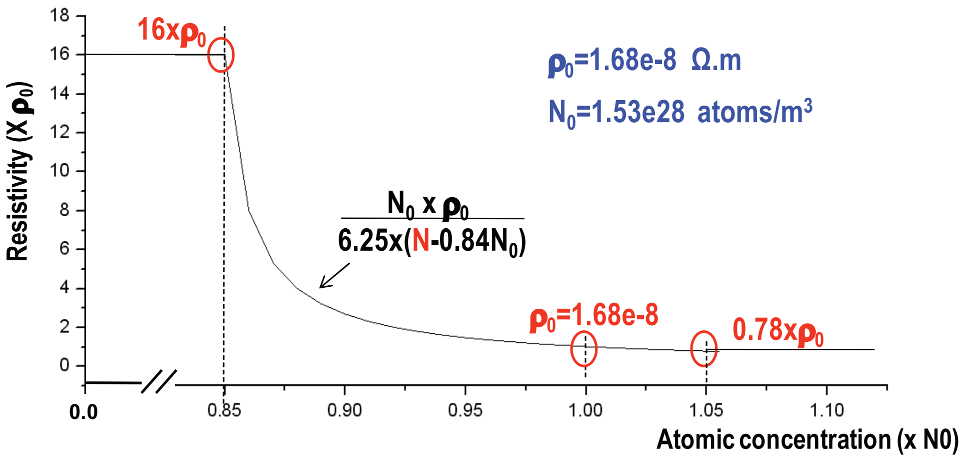


Figure 64: The resistivity function vs atomic concentration.

When N is equal to N_0, ρ is constrained as ρ_0. When N $\leq 85\%N_0$, ρ is saturated at a high value, where we choose the arbitrary large number 16ρ. When $85\%N_0 \leq N \leq N_0$, ρ is reversely dependent to the concentration. When N $\geq N_0$, ρ has very little decrease. The accuracy of this resistivity function is still under investigation.

7.5.2 TSV Resistance Evolution

The simulated TSV effective resistance evolution is plotted in Figure 65, where the TSV has 5um diameter, 25um depth, 1.9um grains, and 0.1um-thick grain boundary thickness, and the wires connecting to the top and bottom of the TSV have 6um width, 10um length, and 1um thickness. At the early time period, the effective resistance of the TSV increases very fast. After 2e9 seconds, the resistance is saturated. Up to 19% TSV resistance increases is observed.

7.5.3 Adding Grains in Wires

In the previous simulations, all wires have no grain, which simulates the EM phenomenon of bamboo wire structures. In this section, the test structure is extended to contain grains in the wires that are the non-bamboo structure as shown in Figure 66. The grains in the wires illustrated in Figure 66(b) have the same grain and grain boundary structure as the TSV, where grain size is 1.9um, grain boundary thickness is 0.1um. In grained wires, the atoms can now transport through the wires also.

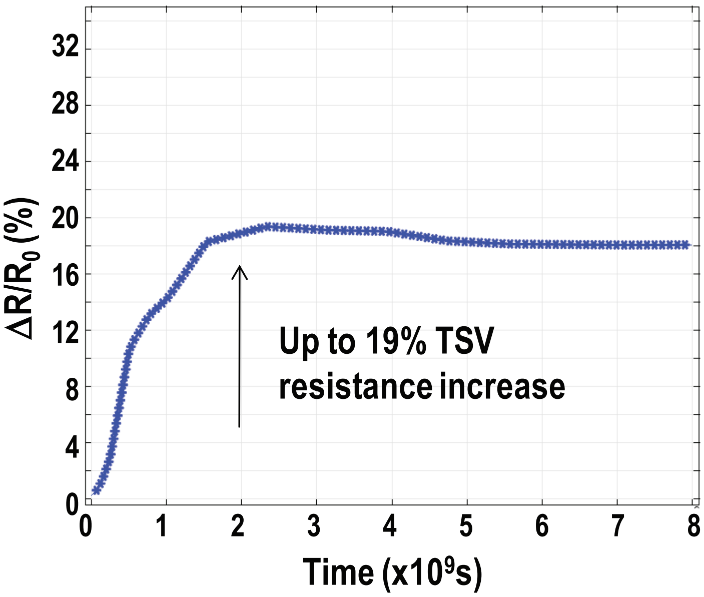


Figure 65: The simulation of TSV effective resistance changes over time.

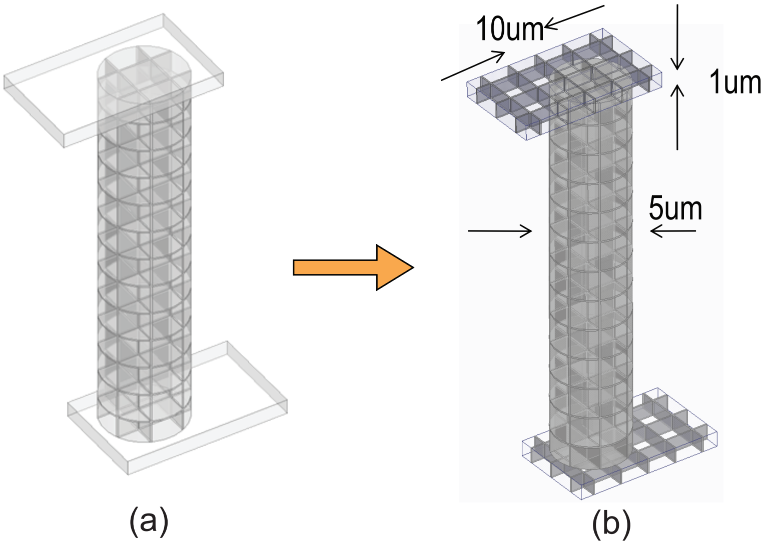


Figure 66: Adding grains in the wires. (a) Bamboo wire with no grains. (b) non-bamboo wires with grains.

The simulated TSV effective resistance is plotted in Figure 67. This resistance has similar trend as the case when no grains are presented in the wires. In addition, the maximum resistance increase can reach to 28% compared with the initial value. This resistance change is greater than the one in Figure 65 when wires are bamboo structure. The major reason is because that the local current density in the wires is higher when wires have grains than the case when wires have no grains. In the case of bamboo wires (no grain in the wires), the current in the wire is uniform before diving into the TSV. However, in the case of grained wires, the current density is nonuniform inside the wires, the local high current density will make the current crowding higher at the TSV-to-wire interfaces.

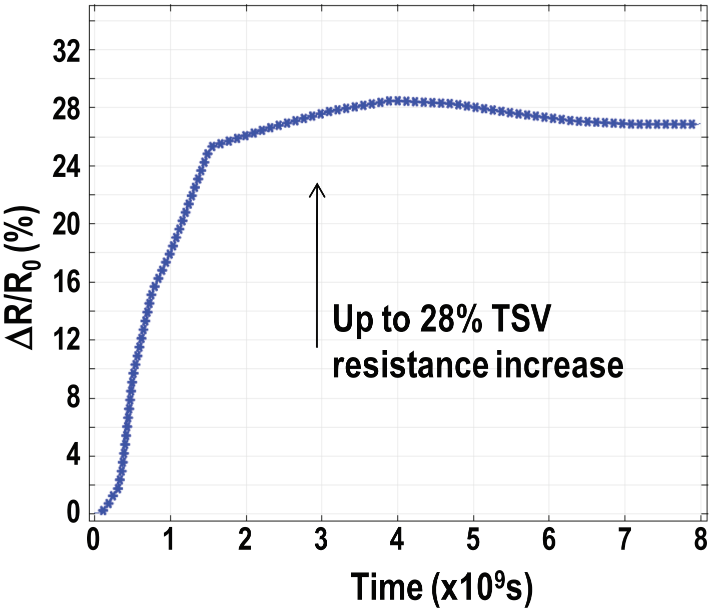


Figure 67: The simulated TSV effective resistance evolution when wires have grains.

The detailed current density distribution is plotted in Figure 68 when time is 31.7 years. First, at the top TSV-to-wire interfaces (plots on the left), the current crowds not only at the TSV-to-wire connection but also along the grain boundaries of the wires. Because the top-left interface has accumulated atoms, which means low resistivity referring to the equations. Since most accumulations occur at the grain boundary, the effective resistance along the grain boundary is lower than that in the grains. Therefore, most current concentrates at the grain boundaries in the wires. Second, with the XY plane moves towards TSV center, the current density becomes uniform. Third, for the bottom-right TSV-to-wire interface, the current tends to crowd again with XY plane moves to the bottom. The current crowds at the TSV-to-wire interfaces. In addition, the current concentrates at the grains of the wires. This is because that the atomic depletion occurs at the bottom-right interface, where the resistivity of the grain boundary is much higher than that of the grains. Therefore, most of the current concentrates at grains instead of grain boundaries in the bottom wires.

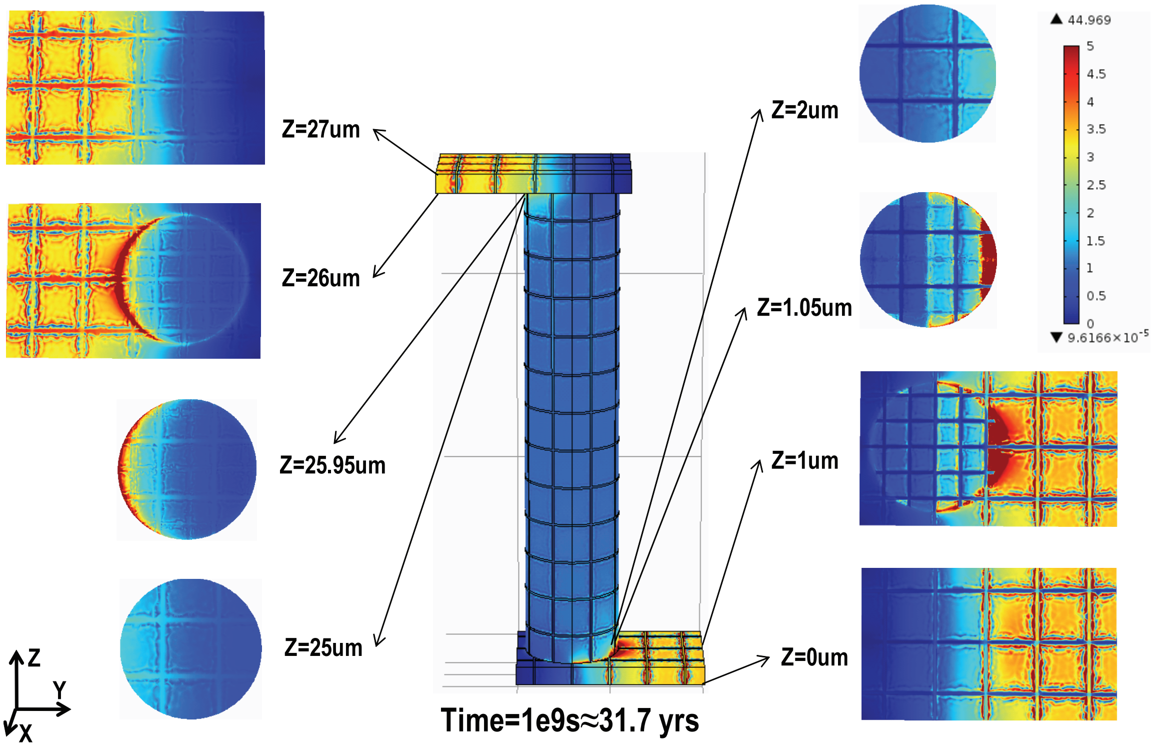


Figure 68: Current density distribution in 3D view and XY planes when wires contain grains. The current density is normalized to the TSV average current density (5mA/um^2).

7.6 Summary

In this chapter, electromigration (EM) has been studied by modeling atomic concentration in TSVs and TSV effective resistance change including the effect of grain boundaries. From a set of extensive investigations, our observations are as follows: (1) Atomic concentration depleted or accumulated at the corner of wire-to-TSV interfaces, where the high current density are crowded; (2) Potential hillocks and voids inside the TSV have been simulated at the corner of wire-to-TSV interfaces; (3) High temperature, large current density, small grain size, or low activation energy of grain boundaries can accelerate the electromigration, thus shortening the lifetime of the TSV. By performing transient analysis and defining the resistivity function, the TSV effective resistance change over time is able to simulated.

CHAPTER VIII

CONCLUSIONS AND FUTURE WORKS

8.1 Conclusions

Three-dimensional integrated circuit (3D IC) has emerged as a promising technology to continue the scaling trajectory predicted by Moores Law for future IC generations. Recent 3D research has focused on improving performance, lowering power consumption, increasing reliability and manufacturability, and designing testing schemes. Reliable clock and power network designs play an important role in pushing the mainstream acceptance of 3D ICs.

This dissertation has addressed many reliability issues in clock and power distribution networks for 3D ICs. For 3D clock synthesis, challenges issues, including pre-bond testability, TSV-induced obstacle avoidance, and TSV array utilization have been taken care. Meanwhile, three important general design goals, including low power, skew, and slew, have been ensured in 3D clock designs.

In addition to reliable clock design, power integrity analysis for EM reliability has also been addressed in this dissertation. Investigations and modeling on current crowding and electromigration for TSV-based 3D connections have been performed.

The following works have been presented in this thesis:

- A comprehensive clock synthesis algorithm for 3D ICs;
- An in-depth investigation on the impact of TSV utilization on 3D clock performance;
- The first clock design methodology for pre-bond testing in 3D ICs;
- The first clock synthesis algorithm for TSV-induced obstacle avoidance;
- The first clock synthesis algorithm of TSV array utilization for low-power 3D clock design;

- A detailed investigation on current density distribution in TSV-to-wire interface and a TSV model for 3D power integrity analysis;
- The first multi-physics modeling approach for transient analysis on Electromigration in TSV-based 3D connections.

First, design optimization techniques for reliable low-power and low-slew 3D clock network design have been investigated. TSV utilization has shown significant impact on clock power consumption: More TSVs helps to reduce the wirelength and power consumption; using TSVs with large parasitic capacitance may increase clock power when too many TSVs are used. Second, to ensure the pre-bond testing, which test each individual die before bonding, the 3D clock design methodology has been developed and implemented. The generated 3D clock network is able to ensure both pre-bond testability and post-bond operation with minimum skew and short wirelength. Third, a practical obstacle issue in TSV-based 3D clock tree synthesis has been studied. The proposed clock routing algorithm can avoid overlapping with TSV-induced obstacles with minimum skew and do not sacrifice wirelength or clock power. Fourth, the proposed decision-tree-based clock synthesis (DTCS) method explores the entire solution space for the best TSV array utilization in terms of low power. Close-to-optimal solutions can be found for power efficiency with skew minimization in short runtime.

Moreover, the current-density distribution inside the 3D TSV-based power grids has been investigated. A large current gradient called current crowding near the interface between power wires and TSVs has been observed. The 3D TSV model has been implemented with a good accuracy and far less complexity compared with the finite-element tools. The proposed simple TSV model has been applied on chip-scale 3D PDNs to analyze detailed current-density distributions and voltage drops. Finally, electromigration (EM) has been studied by modeling atomic concentration in TSVs and simulating the TSV effective resistance, which includes the effect of grain and grain boundary structure. Atomic concentration depleted or accumulated at the corner of wire-to-TSV interfaces, where the high current density are crowded. High temperature, large current density, small grain size, or low activation energy of grain boundaries can accelerate the electromigration, thus shortening the lifetime of the TSV.

8.2 Future Works

Many important reliability issues in clock and power network designs should be addressed in the future: TSV redundancy is important to ensure that the clock signal can be delivered safely when TSV faults present. Each clock TSV can be assigned a redundant TSV right close to it. However, it may occupy significant silicon area and brings in large congestion due to the large-scale 3D clock network. In addition, the clock skew should be taken care when a fault TSV is replaced by a redundant one. Thus, an efficient redundancy for 3D clock network with minimum skew and low power is important.

The TSV coupling, especially in the clock TSVs with high switching activities, should be considered in the 3D clock synthesis. The TSV coupling capacitance is non-negligible in 3D clock network. An investigation on the TSV coupling capacitance on clock timing should be performed. Meanwhile, to reduce the coupling effect, driver sizing or TSV shielding should be performed. Since P/G TSVs can work as shielding TSVs, the co-design of 3D clock and power network and co-utilization of P/G TSVs and clock TSVs can be an interesting research direction.

The EM modeling work is able to help the designers allocate the EM risk locations of the 3D connection and analyze the atomic concentration evolution overtime. These simulations should also allow comparisons with experimental measurements of void and hillock formation and measurements of EM lifetime. The modeling approach can be smoothly extended to include irregular grain structures in the TSVs, grain boundaries in both the wires and TSVs, surface diffusion, grain boundary thickness, grain boundary non-uniformity, and other physical details. Many assumptions have been included in this work. We did not consider physical effect such as grain migration, void nucleation and growth. However, in principle, these phenomena could be integrated in the modeling approach.

REFERENCES

[1] Zhao, X., Mukhopadhyay, S., and Lim, S. K., "Variation-Tolerant and Low-Power Clock Network Design for 3D ICs," in *IEEE Electronic Components and Technology Conf.*, pp. 2007–2014, 2011.

[2] J.U.Knickerbocker, P.S.Andry, and et al., "Three-dimensional Silicon Integration," in *IBM Journal of Research and Development*, vol. 52, pp. 553–569, 2008.

[3] Lim, S. K., "TSV-Aware 3D Physical Design Tool Needs for Faster Mainstream Acceptance of 3D ICs," in *ACM DAC Knowledge Center, http://www.dac.com*, 2010.

[4] Van der Plas*, et al*, G., "Design Issues and Considerations for Low-Cost 3-D TSV IC Technology," *IEEE Journal of Solid-State Circuits*, vol. 46, pp. 293–307, January 2011.

[5] Vardaman, J., "3-D Through-Silicon Vias Become a Reality," 2007. http://www.semiconductor.net/article/CA6445435.html.

[6] "The International Technology Roadmap For Semiconductors." http://www.itrs.net/.

[7] Restle, P. J., McNamara, T. G., Webber, D. A., Camporese, P. J., Eng, K. F., Jenkins, K. A., Allen, D. H., Rohn, M. J., Quaranta, M. P., Boerstler, D. W., Alpert, C. J., Carter, C. A., Bailey, R. N., Petrovick, J. G., Krauter, B. L., and McCredie, B. D., "A Clock Distribution Network for Microprocessors," *Solid-State Circuits, IEEE Journal of*, vol. 36, pp. 792–799, August 2001.

[8] Friedman, E. G., "Clock Distribution Networks in Synchronous Digital Integrated-circuits," *Proceedings of the IEEE*, vol. 89, pp. 665–692, May 2001.

[9] Zhu, Q. K., "High-Speed Clock Network Design," *published by Springer*, 2003.

[10] Lewis, D. L. and Lee, H.-H. S., "A Scan-Island Based Design Enabling Pre-bond Testbility in Die-Stacked Microprocessors," pp. 1–8, 2007.

[11] Black, J. R., "Electromigration–A Brief Survey and Some Recent Results," *Electron Devices, IEEE Transactions on*, vol. 16, pp. 338–347, April 1969.

[12] Abella, J. and Vera, X., "Electromigration for Microarchitects," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, pp. 9:1–9:18, February 2010.

[13] Tu, K. N., "Recent advances on electromigration in very-large-scale-integration of interconnects," *Journal of Applied Physics*, vol. 94, pp. 5451–5473, November 2003.

[14] Pak, J., Pathak, M., Lim, S. K., and Pan, D. Z., "Modeling of Electromigration in Through-Silicon-Via based 3D IC," in *IEEE Electronic Components and Technology Conf.*, pp. 1420–1427, 2011.

[15] Tan, Y. C., Tan, C. M., Zhang, X. W., Chai, T. C., and Yu, D. Q., "Electromigration performance of Through Silicon Via (TSV), A modeling approach," *Microelectronics Reliability*, vol. 50, pp. 1336–1340, September-November 2010.

[16] Bakoglu, H. B., Walker, J. T., and Meindl, J. D., "A Symmetric Clock-Distribution Tree and Optimized High-Speed Interconnections for Reduced Clock Skew in ULSI and WSI Circuits," in *Proc. IEEE Int. Conf. on Computer Design*, pp. 118–122, 1986.

[17] Wann, D. F. and Franklin, M. A., "Asynchronous and Clocked Structures for VLSI Based Interconnection Networks," *IEEE Transactions on Computers*, vol. 21, pp. 284–193, March 1983.

[18] Jackson, M., Srinivasan, A., and Kuh, E., "Clock Routing for High-Performance ICs," in *Proc. ACM Design Automation Conf.*, pp. 573–579, 1990.

[19] Cong, J., Kahng, A., and Robins, G., "Matching-based methods for high-performance clock routing," *Proc. IEEE Int. Conf. on Computer-Aided Design*, vol. 12, pp. 1157–1169, August 1993.

[20] Tsay, R.-S., "An Exact Zero-Skew Clock Routing Algorithm," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, pp. 242–249, February 1993.

[21] Elmore, W. C., "The Transient Analysis of Damped Linear Networks with Particular Regard to Wideband Amplifiers," *Journal of Applied Physics*, vol. 19, pp. 55–63, July 1948.

[22] Chao, T.-H., Hsu, Y.-C., Ho, J.-M., and Kahng, A., "Zero Skew Clock Routing with Minimum Wirelength," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 39, pp. 799–814, November 1992.

[23] Kuhn, K. *et al.*, "Managing Process Variation in Intels 45nm CMOS Technology," *Intel Technology Journal*, vol. 12, pp. 92–110, June 2008.

[24] Sauter, S., Schmitt-Landsiedel, D., Thewes, R., and Weber, W., "Effect of Parameter Variations at Chip and Wafer Level on Clock Skews," *IEEE Trans on Semiconductor Manufacturing*, vol. 13, pp. 395 –400, November 2000.

[25] Narasimhan, A. and Sridhar, R., "Impact of Variability on Clock Skew in H-tree Clock Networks," in *International Symposium on Quality Electronic Design*, pp. 458–466, 2007.

[26] Bowman, K., Alameldeen, A., Srinivasan, S., and Wilkerson, C., "Impact of Die-to-Die and Within-Die Parameter Variations on the Clock Frequency and Throughput of Multi-Core Processors," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 17, pp. 1679–1690, December 2009.

[27] Neves, J. and Friedman, E., "Design Methodology for Synthesizing Clock Distribution Networks Exploiting Nonzero Localized Clock Skew," *IEEE Trans. on VLSI Systems*, vol. 4, pp. 286–291, June 1996.

[28] Padmanabhan, U., Wang, J., and Hu, J., "Robust Clock Tree Routing in the Presence of Process Variations," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, pp. 1385–1397, August 2008.

[29] Lam, W.-C. and Koh, C.-K., "Process Variation Robust Clock Tree Routing," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 606–611, 2005.

[30] Venkataraman, G., Sze, C., and Hu, J., "Skew Scheduling and Clock Routing for Improved Tolerance to Process Variations," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 594–599, 2005.

[31] Rajaram, A., Hu, J., and Mahapatra, R., "Reducing Clock Skew Variability via Crosslinks," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, pp. 1176–1182, June 2006.

[32] van Ginneken, L., "Buffer Placement in Distributed RC-tree Networks for Minimal Elmore Delay," in *IEEE International Symposium on Circuits and Systems*, pp. 865–868, 1990.

[33] Lillis, J., Cheng, C.-K., and Lin, T.-T., "Optimal Wire Sizing and Buffer Insertion for Low Power and A Generalized Delay Model," *IEEE Journal of Solid-State Circuits*, vol. 31, pp. 437–447, March 1996.

[34] Tellez, G. E. and Sarrafzadeh, M., "Minimal Buffer Insertion in Clock Trees with Skew and Slew Rate Constraints," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, pp. 333–342, April 1997.

[35] Albrecht, C., Kahng, A. B., Liu, B., Mandoiu, I. I., and Zelikovsky, A. Z., "On the Skew-Bounded Minimum-Buffer Routing Tree Problem," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, pp. 937–945, July 2003.

[36] Alpert, C. J., Kahng, A. B., Liu, B., Mandoiu, I. I., and Zelikovsky, A. Z., "Minimum Buffered Routing with Bounded Capacitive Load for Slew Rate and Reliability Control," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, pp. 241–253, March 2003.

[37] Hu, S., Alpert, C. J., Hu, J., Karandikar, S. K., Li, Z., Shi, W., and Sze, C. N., "Fast Algorithms for Slew-Constrained Minimum Cost Buffering," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, pp. 2009–2022, November 2007.

[38] Wang, K., Ran, Y., Jiang, H., and Marek-Sadowska, M., "General skew constrained clock network sizing based on sequential linear programming," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, pp. 773–782, May 2005.

[39] Guthaus, M. R., Sylvester, D., and Brown, R. B., "Clock Buffer And Wire Sizing using Sequential Programming," in *Proc. ACM Design Automation Conf.*, pp. 1041–1046, 2006.

[40] Cho, M., Ahmed, S., and Pan, D. Z., "TACO: Temperature aware clock-tree optimization," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, pp. 581–586, 2005.

[41] Chakraborty, A., Sithambaram, P., Duraisami, K., Macii, A., and Poncino, M., "Thermal resilient bounded-skew clock tree optimization methodology," in *Proc. Design, Automation and Test in Europe*, pp. 832–837, 2006.

[42] Yu, H., Hu, Y., Liu, C., and He, L., "Minimal skew clock embedding considering time variant temperature gradient," in *Proc. Int. Symp. on Physical Design*, pp. 173–180, 2007.

[43] Pavlidis, V. F., Savidis, I., and Friedman, E. G., "Clock Distribution Networks for 3-D Integrated Circuits," in *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 651–654, 2008.

[44] Arunachalam, V. and Burleson, W., "Low-Power Clock Distribution in A Multilayer Core 3D Microprocessor," in *Proc. Great Lakes Symposum on VLSI*, pp. 429–434, 2008.

[45] Minz, J., Zhao, X., and Lim, S. K., "Buffered Clock Tree Synthesis for 3D ICs Under Thermal Variations," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 504–509, 2008.

[46] Xu, H., Pavlidis, V. F., and De Micheli, G., "Process-induced Skew Variation for Scaled 2-D and 3-D ICs," in *Proc of International workshop on System level Interconnect Prediction*, pp. 17–24, 2010.

[47] Lee, H.-H. S. and Chakrabarty, K., "Test Challenges for 3D Integrated Circuits," vol. 26, pp. 26–35, Septemeber-October 2009.

[48] Marinissen, E. J. and Zorian, Y., "Testing 3D Chips Containing Through-Silicon Vias," pp. 1–11, 2009.

[49] Lewis, D. L. and Lee, H.-H. S., "Testing Circuit-Partitioned 3D IC Designs," pp. 139–144, 2009.

[50] Jiang, L., Huang, L., and Xu, Q., "Test Architecture Design and Optimization for Three-Dimensional SoCs," in *Proc. Design, Automation and Test in Europe*, pp. 220–225, 2009.

[51] Jiang, L., Xu, Q., Chakrabarty, K., and Mak, T. M., "Layout-Driven Test-Architecture Design and Optimization for 3D SoCs under Pre-Bond Test-Pin-Count Constraint," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, pp. 191–196, 2009.

[52] Kahng, A. B. and Tsao, C.-W. A., "More Practical Bounded-Skew Clock Routing," in *Proc. ACM Design Automation Conf.*, pp. 594–599, 1997.

[53] Kim, H. and Zhou, D., "Efficient Implementation of a Planar Clock Routing with the Treatment of Obstacles," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 10, pp. 1220–1225, 2000.

[54] Huang, H., Luk, W.-S., Zhao, W., and Zeng, X., "DME-Based Clock Routing in the Presence of Obstacles," in *in Proceedings of 7th International Conference on ASIC*, pp. 429–434, 2008.

[55] Liu, W.-H., Li, Y.-L., and Chen, H.-C., "Minimizing Clock Latency Range in Robust Clock Tree Synthesis," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 389–394, 2010.

[56] Lu, J., Chow, W.-K., Sham, C.-W., and Young, E. F., "A Dual-MST Approach for Clock Network Synthesis," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 467–473, 2010.

[57] Shih, X.-W., Cheng, C.-C., Ho, Y.-K., and Chang, Y.-W., "Blockage-Avoiding Buffered Clock-Tree Synthesis for Clock Latency-Range and Skew Minimization," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 395–400, 2010.

[58] Lau, J. H., "TSV Manufacturing Yield and Hidden Costs for 3D IC Integration," in *IEEE Electronic Components and Technology Conf.*, pp. 1031–1042, 2010.

[59] Mercha*, et al*, A., "Comprehensive Analysis of the Impact of Single and Arrays of Through Silicon Vias Induced Stress on High-k / Metal Gate CMOS Performance," in *IEEE International Electron Devices Meeting*, pp. 2.2.1–2.2.4, 2010.

[60] Song*, et al*, T., "Analysis of TSV-to-TSV Coupling with High-Impedance Termination in 3D ICs," in *Proc. Int. Symp. on Quality Electronic Design*, pp. 122–128, 2011.

[61] Liu, C., Song, T., and Lim, S. K., "Signal Integrity Analysis and Optimization for 3D ICs," in *Proc. Int. Symp. on Quality Electronic Design*, pp. 42–49, 2011.

[62] Yang, J.-S., Athikulwongse, K., Lee, Y.-J., Lim, S. K., and Pan, D. Z., "TSV Stress Aware Timing Analysis with Applications to 3D-IC Layout Optimization," in *Proc. ACM Design Automation Conf.*, 2010.

[63] Kim, D. H., Athikulwongse, K., and Lim, S. K., "A Study of Through-Silicon-Via Impact on the 3D Stacked IC Layout," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, pp. 674 –680, 2009.

[64] Pathak, M., Lee, Y.-J., Moon, T., and Lim, S. K., "Through-Silicon-Via Management during 3D Physical Design: When to Add and How Many?," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, pp. 387 –394, 2010.

[65] Savidis, I. and Friedman, E. G., "Closed-Form Expressions of 3-D Via Resistance, Inductance, and Capacitance," *Electron Devices, IEEE Transactions on*, vol. 56, pp. 1873 –1881, September 2009.

[66] Athikulwongse, K., Chakraborty, A., Yang, J.-S., Pan, D. Z., and Lim, S. K., "Stress-Driven 3D-IC Placement with TSV Keep-Out Zone and Regularity Study," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, pp. 669–674, 2010.

[67] Jung, M., Mitra, J., Pan, D., and Lim, S. K., "TSV Stress-aware Full-Chip Mechanical Reliability Analysis and Optimization for 3D IC," in *Proc. ACM Design Automation Conf.*, pp. 188–193, 2011.

[68] Black, J. R., "Electromigration – A Brief Survey and Some Recent Restuls," *IEEE Transactions on Electron Devices*, vol. ED-16, pp. 338–347, April 1969.

[69] Tu, K. N., "Recent advances on electromigration in very-large-scale-integration of interconnects," *Journal of applied physics*, vol. 94, pp. 5451–5473, Nov. 2003.

[70] Zhao, X., Scheuermann, M., and Lim, S. K., "Analysis of DC Current Crowding in Through-Silicon-Vias and Its Impact on Power Integrity in 3D ICs," in *Proc. ACM Design Automation Conf.*, pp. 157–162, 2012.

[71] Ryu, S.-K., Lu, K.-H., Zhang, X., Im, J.-H., Ho, P., and Huang, R., "Impact of Near-Surface Thermal Stresses on Interfacial Reliability of Through-Silicon Vias for 3-D Interconnects," *IEEE Transactions on Device and Materials Reliability*, vol. 11, pp. 35 –43, March 2011.

[72] Pathak, M., Pak, J., Pan, D. Z., and Lim, S. K., "Electromigration Modeling and Full-chip Reliability Analysis for BEOL Interconnect in TSV-based 3D ICs," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, pp. 555–562, 2011.

[73] Li, W. and Tan, C. M., "Enhanced finite element modelling of Cu electromigration using ANSYS and matlab," *Microelectronics Reliability*, vol. 47, pp. 1497–1501, August 2007.

[74] Cacho, F., Fiori, V., Chappaz, C., Tavernier, C., and Jaouen, H., "Modeling of Electromigration Induced Failure Mechanism in Semiconductor Devices," in *in Proceedings of the COMSOL Users Conference*, pp. 1–6, 2007.

[75] Khan, N., Alam, S., and Hassoun, S., "Power Delivery Design for 3-D ICs Using Different Through-Silicon Via (TSV) Technologies," *IEEE Trans. on VLSI Systems*, vol. 19, pp. 647–658, April 2011.

[76] Zhao, X., Lewis, D. L., Lee, H.-H., and Lim, S. K., "Pre-bond Testable Low-Power Clock Tree Design for 3D Stacked ICs," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, pp. 184–190, 2009.

[77] Zhao, X. and Lim, S. K., "Power and Slew-aware Clock Network Design for Through-Silicon-Via (TSV) Based 3D ICs," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 175–180, 2010.

[78] Kim, T.-Y. and Kim, T., "Clock Tree Embedding for 3D ICs," in *Proc. Asia and South Pacific Design Automation Conf.*, pp. 486–491, 2010.

[79] Katti, G., Stucchi, M., De Meyer, K., and Dehaene, W., "Electrical Modeling and Characterization of Through Silicon Via for Three-Dimensional ICs," *Electron Devices, IEEE Transactions on*, vol. 57, pp. 256 –262, January 2010.

[80] Bandyopadhyay, T., Chatterjee, R., Chung, D., Swaminathan, M., and Tummala, R., "Electrical Modeling of Through Silicon and Package Vias," in *3D System Integration, 2009. 3DIC 2009. IEEE International Conference on*, pp. 1–8, September 2009.

[81] Weerasekera, R., Grange, M., Pamunuwa, D., Tenhunen, H., and Zheng, L.-R., "Compact Modeling of Through-Silicon Vias (TSVs) in Three-Dimensional (3-D) Integrated Circuits," in *3D System Integration, 2009. 3DIC 2009. IEEE International Conference on*, pp. 1 –8, 2009.

[82] "Predictive technology model." http://ptm.asu.edu/.

[83] GSRC Benchmark, http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/BST.

[84] Verigy V93000 SOC Series Pin Scale Digital Cards, http://www1.verigy.com.

[85] RMST-Pack, "The rectilinear minimum spanning tree pack." http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/RSMT/RMST/.

[86] ISPD Contest 2009, http://www.sigda.org/ispd/contests/ispd09cts.html.

[87] Synopsys, Raphael, http://www.synopsys.com.

[88] IWLS2005 Benchmark. http://www.iwls.org/iwls2005/.

[89] Lung, C.-L., Su, Y.-S., Huang, S.-H., Shi, Y., and Chang, S.-C., "Fault-Tolerant 3D Clock Network," in *Proc. ACM Design Automation Conf.*, pp. 645–651, 2011.

[90] Boese, K. D. and Kahng, A. B., "Zero-Skew Clock Routing Trees with Minimum Wirelength," in *Proc. IEEE Intl. Conf. on ASIC*, pp. 1.1.1–1.1.5, 1992.

[91] Extractor, A. Q.

[92] Karmalkar, S., Mohan, P., Nair, H., and Yeluri, R., "Compact Models of Spreading Resistances for ElectricalThermal Design of Devices and ICs," *Electron Devices, IEEE Transactions on*, vol. 54, pp. 1734 –1743, July 2007.

[93] Healy, M. B. and Lim, S. K., "Distributed TSV Topology for 3-D Power-Supply Networks," *IEEE Trans. on VLSI Systems*, vol. PP, pp. 1–14, October 2011.

[94] Li, W. and Tan, C. M., "Enhanced finite element modelling of Cu electromigration using ANSYS and matlab," *Microelectronics Reliability*, vol. 47, no. 9-11, pp. 1497–1501, 2007.

[95] COMSOL, http://www.comsol.com/.

[96] Tu, K. N., "Electromigration in Stressed Thin Films," *Phys. Rev. B*, vol. 45, pp. 1409–1413, January 1992.

[97] Blech, I. A., "Diffusional back flows during electromigration," *Acta Materialia*, vol. 46, pp. 3717–3723, July 1998.

[98] Hu, C.-K., Gignac, L. M., Liniger, E., Huang, E., Greco, S., McLaughlin, P., Yang, C.-C., and Demarest, J. J., "Electromigration Challenges for Nanoscale Cu Wiring," *AIP Conference Proceedings*, vol. 1143, no. 1, pp. 3–11, 2009.

PUBLICATIONS

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

[1] **Xin Zhao**, Jacob Minz, and Sung Kyu Lim, "Low-Power and Reliable Clock Network Design for Through Silicon Via based 3D ICs", in *IEEE Transactions on Components, Packaging and Manufacturing Technology*, Vol. 1, No. 2, pp. 247-259, 2011.

[2] **Xin Zhao**, Dean L. Lewis, Hsien-Hsin S. Lee, and Sung Kyu Lim, "Low-Power Clock Tree Design for Pre-Bond Testing of 3D Stacked ICs", in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 30, No. 5, pp. 732-745, 2011. (Nominated as Best Paper)

[3] **Xin Zhao** and Sung Kyu Lim, "TSV Array Utilization in Low-Power 3D Clock Network Design", in *IEEE International Symposium on Low Power Electronics and Design*, pp. 21-26, 2012. (Nominated as Best Paper)

[4] **Xin Zhao**, Michael Scheuermann, and Sung Kyu Lim, "Analysis of DC Current Crowding in Through-Silicon-Vias and Its Impact on Power Integrity in 3D ICs", in *ACM Design Automation Conference*, pp. 157-161, 2012.

[5] **Xin Zhao** and Sung Kyu Lim, "Through-Silicon-Via-Induced Obstacle-Aware Clock Tree Synthesis for 3D ICs", in *IEEE/ACM Asia South Pacific Design Automation Conference*, pp.347-352, 2012.

[6] **Xin Zhao** and Sung Kyu Lim, "Power and Slew-aware Clock Network Design for Through-Silicon-Via (TSV) Based 3D ICs", in *IEEE/ACM Asia South Pacific Design Automation Conference*, pp 175-180, 2010.

[7] **Xin Zhao**, Dean Lewis, Hsien-Hsin S. Lee, and Sung Kyu Lim, "Pre-bond Testable Low-Power Clock Tree Design for 3D Stacked ICs", in *IEEE International Conference*

on Computer-Aided Design, pp.184-190, 2009. (Nominated as Best Paper)

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

[8] Kwanyeob Chae, **Xin Zhao**, Sung Kyu Lim, and Saibal Mukhopadhyay, "A Post-Silicon Tuning Method to Minimize Clock Skew Variations in 3D ICs", in *IEEE Transactions on Components, Packaging and Manufacturing Technology*, (submitted)

[9] **Xin Zhao**, Jeremy R. Tolbert, Chang Liu, Saibal Mukhopadhyay, and Sung Kyu Lim, "Variation-aware Clock Network Design Methodology for Ultra-Low Voltage (ULV) Circuits", in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 31, No. 8, pp. 1222-1234, 2012.

[10] Jeremy R. Tolbert, **Xin Zhao**, Sung Kyu Lim, and Saibal Mukhopadhyay, "Analysis and Design of Energy and Slew Aware Subthreshold Clock Systems", in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 30, No. 9, pp. 1349-1358, 2011.

[11] Yang Shang, Chun Zhang, Hao Yu, **Xin Zhao**, and Sung Kyu Lim, "Thermal-reliable 3D Clock-tree Synthesis Considering Nonlinear Electrical-thermal-coupled TSV Model", in *IEEE/ACM Asia South Pacific Design Automation Conference*, 2013.

[12] Kwanyeob Chae, **Xin Zhao**, Amit R. Trivedi, Sung Kyu Lim, and Saibal Mukhopadhyay, "Post-Silicon Tuning Method for 3D Clock Network To Minimize Clock Skews", in *SRC TECHCON Conference*, 2012.

[13] Dae Hyun Kim, Krit Athikulwongse, Michael B. Healy, Mohammad M. Hossain, Moongon Jung, Ilya Khorosh, Gokul Kumar, Young-Joon Lee, Dean L. Lewis, Tzu-Wei Lin, Chang Liu, Shreepad Panth, Mohit Pathak, Minzhen Ren, Guanhao Shen, Taigon Song, Dong Hyuk Woo, **Xin Zhao**, Joungho Kim, Ho Choi, Gabriel H. Loh, Hsien-Hsin S. Lee, and Sung Kyu Lim, "3D-MAPS: 3D Massively Parallel Processor

with Stacked Memory", in *IEEE International Solid-State Circuits Conference*, pp. 188-189, 2012.

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

[15] **Xin Zhao**, Jeremy Tolbert, Chang Liu, Saibal Mukhopadhyay, and Sung Kyu Lim, "Variation-aware Clock Network Design Methodology for Ultra-Low Voltage (ULV) Circuits", in *IEEE International Symposium on Low Power Electronics and Design*, pp. 9-14, 2011.

[16] **Xin Zhao**, Saibal Mukhopadhyay, and Sung Kyu Lim, "Variation-Tolerant and Low-Power Clock Network Design for 3D ICs", in *IEEE Electronic Components and Technology Conference*, pp. 2007-2014, 2011.

[17] Jae-Seok Yang, Jiwoo Park, **Xin Zhao**, Sung Kyu Lim, and David Pan, "Robust Clock Tree Synthesis with Timing Yield Optimization for 3D ICs", in *IEEE/ACM Asia South Pacific Design Automation Conference*, pp.621-626, 2011.

[18] Michael B. Healy, Krit Athikulwongse, Rohan Goel, Mohammad M. Hossain, Dae Hyun Kim, Young-Joon Lee, Dean L. Lewis, Tzu-Wei Lin, Chang Liu, Moongon Jung, Brian Ouellette, Mohit Pathak, Hemant Sane, Guanhao Shen, Dong Hyuk Woo, **Xin Zhao**, Gabriel H. Loh, Hsien-Hsin S. Lee, and Sung Kyu Lim, "Design and Analysis of 3D-MAPS: A Many-Core 3D Processor with Stacked Memory", in *IEEE Custom Integrated Circuits Conference*, pp.1-4, 2010.

[19] Krit Athikulwongse, **Xin Zhao**, and Sung Kyu Lim, "Buffered Clock Tree Sizing for Skew Minimization Under Power and Thermal Budgets", in *IEEE/ACM Asia South Pacific Design Automation Conference*, pp.474-479, 2010.

[20] Jeremy Tolbert, **Xin Zhao**, Saibal Mukhopadhyay, and Sung Kyu Lim, "Slew-Aware Clock Tree Design For Reliable Subthreshold Circuits", in *IEEE International Symposium on Low Power Electronics and Design*, pp. 15-20, 2009.

[21] Jacob Minz, **Xin Zhao**, and Sung Kyu Lim, "Buffered Clock Tree Synthesis for 3D ICs Under Thermal Variations", in *IEEE/ACM Asia South Pacific Design Automation Conference, pp.504-509*, 2008.

[22] Mongkol Ekpanyapong, **Xin Zhao**, Sung Kyu Lim, "An Efficient Computation of Statistically Critical Sequential Paths Under Retiming," in *IEEE/ACM Asia South Pacific Design Automation Conference*, pp. 547-552, 2007.

VITA

Xin Zhao was born in Beijing, China, 1981. She received the B.S. degree in Electronic Engineering in 2003 and the M.S. degree in Computer Science and Technology in 2006 both from Tsinghua University, Beijing, China. She is currently a Ph.D. candidate in Georgia Institute of Technology.

From 2003 to 2006, she worked as a research assistant at EDA lab in Tsinghua University performing research in the area of physical design for high-performance VLSI design, guided by Professor Yici Cai. From 2007 to present, she has been a graduate research assistant in Georgia Tech Computer-Aided Design (GTCAD) laboratory conducted by Professor Sung kyu Lim. She has been working in the areas of reliable clock delivery network design, power delivery network analysis, and reliability modeling and simulation for 3D VLSI design and low-power VLSI designs.

She has received best paper nominations in IEEE International Conference on Computer-Aided Design (ICCAD) 2009, IEEE Transactions on Computer-Aided Design (TCAD) 2012, and IEEE International Symposium on Low Power Electronics and Design (ISLPED) 2012. She has also received the Outstanding Master Thesis Awards from Tsinghua Univerisy.