Dynamic Through Silicon Via Clustering in 3D IC Floorplanning for Early Performance Optimization

by

Sucheta Mohapatra

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
in
Electrical and Computer Engineering

Dissertation Committee:
Malgorzata Chrzanowska-Jeske, Chair
Richard Campbell
Dan Hammerstrom
Bart Massey

Portland State University
2020

Abstract

Through-silicon via (TSV)-based three-dimensional integrated circuits (3D ICs) are expected to be the breakthrough technology for keeping up with the scaling trends of Moore's law, while also offering the unique opportunity for functional diversification through heterogenous integration. TSVs are vertical metal interconnects enabling communication across stacked and thinned dies. The dramatic reduction in global wirelength and chip footprint in 3D ICs, directly improves delay, device density, bandwidth and routing congestion. Even with the current maturation of TSV process, the roadmap for industry adoption of 3D ICs remains largely uncertain due to lack of standardized 3D tools capable of handling the sheer complexity of the three-dimensional solution space.

Many critical design issues arise due to usage of TSVs. Large-sized TSVs, introduce significant area and delay overhead. The increased risk of TSV failure during fabrication or bonding, causes long-term reliability issues and loss of yield. The earlier these critical issues are addressed in the design cycle, the better our chances are of making realistic performance predictions and informed decisions, for speeding-up convergence. 3D IC floorplanning constitutes an important first step of layout design, providing early feedback on critical performance metrics, i.e., area, wirelength, delay, power and wiring density. Since the resulting floorplan impacts the optimization of all subsequent stages, there is a critical need for efficient TSV-aware layout design exploration tools, which can accurately characterize the physical and electrical impact of TSVs.

A key concept of this thesis is that interconnect delay and power of 3D chips is directly controlled by the quality of the generated 3D floorplan, which is fundamentally impacted by the heuristics guiding the search and evaluation of floorplan. In support of this view, the core objective of this thesis is to develop an efficient methodology to improve the 3D floorplan solution quality. By generating more realistic 3D layouts, we seek to improve the accuracy of evaluation of the goodness of a 3D floorplan. A new dynamic TSV clustering algorithm is introduced, which simultaneously optimizes the sizes and positions of TSV clusters on the layout. This is the first work to consider the direct minimization of TSV occupied area at the floorplanning stage. As the generated floorplan is independent of any fixed arrangement of TSVs as input, it facilitates a more realistic and accurate evaluation of floorplan metrics. A novel nets-to-TSVs assignment algorithm which considers the inherent trade-off between TSV area and the TSV capacitance during net delay optimization, is also included. Experimental results with GSRC benchmarks show average 25% reduction in TSV footprint for all benchmarks, as compared to the single TSV placement approach. Compared to floorplanning with fixed-sized TSV islands, the approach reduces total chip area by average 7.6% and total interconnect delay by average 9%.

In this thesis, early estimation of buffers is directly incorporated with assignment of nets to TSVs. The candidate location of buffers in individual 3D nets is estimated by simultaneously considering the TSVs' RC parasitics, positions of TSVs along the 3D net and size of the cluster containing the TSVs. This results in a more reliable estimate of buffers, interconnect delay and power. Secondly, an analytical approach for estimating the optimal position of buffers around TSVs is developed, which helps in avoiding excessive usage of buffers around TSVs.

The extent to which chip performance is influenced due to negative impact of TSVs, is also determined by the process technology used to fabricate 3D ICs. This important design concern is addressed, by including the impact of future nano-CMOS technologies on early estimation of area, number of buffers, total delay and power. Further insight is gained on the impact of nano-scale TSVs on design quality, when combined with different nanometer technologies.

A new TSV redundancy scheme is incorporated in the floorplanner, for increasing the fault-tolerance of TSV clusters and improving the overall reliability of the design. Assuming a uniform TSV failure rate and an independent TSV defect distribution, a minimum required number of spare TSVs are allocated for the given size of TSV cluster, such that a cluster is fully repairable. Unlike previous works on TSV redundancy based on fixed layouts, the proposed scheme does not incur additional TSV area overhead due to allocation of spare TSVs.

This work addresses some of the most practically relevant challenges in the realm of 3D IC design. The author hopes that the findings in this work provide a basis for formulating guidelines for high-quality flows for 3D design automation.

Dedication

I was taught that the way of progress was neither swift nor easy.

-Marie Curie (French-Polish physicist)

This dissertation is dedicated to

My beloved parents and my brother, who have relentlessly encouraged me to persevere and strive for excellence in all my endeavors.

My late mother-in-law Mrs. Malati Mohapatra, whose kind soul always inspires me to embrace life's challenges with grace, optimism and boundless faith in God.

Acknowledgments

This work is the culmination of a memorable journey of six years of my professional life. I am filled with gratitude for my research advisor Prof. Malgorzata Chrzanowska-Jeske for giving me a platform for research at the Design Automation for VLSI & Emerging Technologies Laboratory at the Portland State University. Her wealth of knowledge and effective mentoring fueled my interest in exploring new solutions for some of the most practical yet challenging problems in design automation, particularly in 3D integration. Throughout my stint as a graduate student, I have learned a lot from her insights on technical writing, critical analysis, taking initiative and ownership of ideas. I am also thankful for her constant support when I was trying to balance my research work while also being a new mom.

I am deeply obliged to my committee members – Prof. Richard Campbell, Prof. Dan Hammerstrom and Prof. Bart Massey for their valuable time and effort in evaluating my thesis. I will always remember their enlightening discussion and encouragement, before my Ph.D. defense. I had the wonderful opportunity to collaborate with my friends and fellow lab-members - Mohd. Abrar Ahmed and Satya K. Vendra. I greatly appreciate their helpful discussions and sharing valuable feedback on my research ideas, particularly on topic of 3D floorplan evaluation.

This journey would have been impossible without the unbounded love and support of my parents, Smt. Sabita Mohapatra and Shri.Kailash Chandra Mohapatra. I thank my father-in-law Shri. Subash Chandra Mohapatra, who never lets me lose sight of my goals. I thank my brother Shakti Prasanna Mohapatra, for always believing in me and inspiring me to think deeply and be patient. To Jolly, who is always there to spread cheer and lend a listening ear to me. I am grateful to my husband Dr. Chandra Shekhar Mohapatra, who has been my anchor and my source of motivation all these years. I also thank my son Kaustav, for livening up our days with his unlimited energy and smiles. Finally, I am lucky and forever grateful to be surrounded by friends, who are my second family in USA – Anuja, Prabhanshu, Kavya, Praveen, Tarun, Sonal, Mitrabhanu, Tripti, Alka, Vikas, Priyanka and Raj. I deeply appreciate them, for often going out of their ways, to help me in every possible way.

Table of Contents

Abstract i

Dedication iv

Acknowledgments v

List of Tables x

List of Figures xii

Glossary xv

List of Symbols xvi

1. INTRODUCTION 1

1.1. Through-Silicon Via (TSV) based Three-dimensional Integrated Circuits 2

1.1.1. Benefits of TSV-based 3D ICs 3

1.1.2. Motivation 4

1.1.3. Broad Overview of Thesis 5

1.2. Thesis Contributions 6

1.3. Through-Silicon Via Process and Fabrication 10

1.4. Classification of 3D ICs 12

1.5. Challenges in TSV-based 3D ICs 14

1.6. Thesis Organization 17

2. STATE-OF-THE-ART IN 3D FLOORPLANNING 19

2.1. Need for Early 3D Layout Design Exploration 19

2.2. 3D Floorplanning Problem Formulation 20

2.3. 3D Floorplanning: Classification and Representation 21

2.3.1. Floorplan Structures 21

2.3.2. Floorplan Representations 22

2.3.2.1. Floorplan Representations for 3D ICs 23

2.3.2.2. Sequence Pair (SP) Representation for 3D Floorplan 23

2.4. Existing Approaches for 3D Floorplanning 24

2.5. Methodology of Data Generation 26

2.5.1. General Assumptions of 3D Floorplanning 27

2.5.2. Evolutionary Algorithm based 3D Floorplanning Flow 30

3. DYNAMIC THROUGH-SILICON VIA CLUSTERING IN 3D IC FLOORPLANNING 33

3.1. Previous Works on TSV-Aware 3D IC Floorplanning 33

3.2. Chapter Contributions 35

3.3. Design Overhead of TSVs 36

3.4. Delay Estimation in 3D Interconnects 38

3.4.1. Electrical Characterization of TSVs 40

3.4.2. Assumption of Worst-Case TSV Capacitance 41

3.4.3 Consideration of TSV Edge-Effects 42

3.5. Methodology of Dynamic TSV Clustering 44
3.5.1 Moves for Dynamic TSV clustering 46
3.5.2. Probabilistic Selection of Clustering Moves 50
3.5.3. Steps of Dynamic TSV Clustering Algorithm 52
3.6 Design Flow of 3D Floorplanning with Dynamic TSV Clustering (3D-DTC) 54
3.7. Experimental Results 59
3.7.1. Optimization of TSV Cluster Sizes and Floorplan Packing Area 60
3.7.2. Comparison of Area Efficiency with Other Approaches 64
3.7.3. Impact of TSV Clustering without KOZ Sharing 64
3.7.4. Wirelength-aware 3D floorplanning with TSV Clustering 66
3.7.5. Delay-Aware 3D Floorplanning with TSV Clustering 67
3.7.6. Comparison of Runtime with Fixed-sized TSV islands Approach 68
3.7.7. Impact of Probability of Selection of Clustering Moves 69
3.8. Chapter Conclusions 71

4. EARLY BUFFER COUNT ESTIMATION IN 3D ICs 72
4.1. Challenges of Early Buffer Estimation in 3D ICs 73
4.2. Previous Related Works 74
4.3. Estimation of Buffer Insertion Length in 3D Interconnects 77
4.4. Influence of TSV Cluster Size on Buffer Estimation 83
4.5. General Flow of 3D Floorplanning with Buffer Estimation 84
4.6. Experimental Results 85
4.6.1. Comparison with Previous Approaches [7][8] 86
4.6.2. Impact of Process Technology on Buffer Insertion Length 89
4.6.3. Impact of Nanoscale TSVs on Buffer insertion Length 90
4.6.4. Impact of TSV Downscaling on Performance in Buffered 3D ICs 91
4.6.5. Impact of TSV Cluster Size on Optimal Buffer Location 92
4.7. Chapter Conclusions 93

5. IMPACT OF TSVS ON PERFORMANCE IN NANO-SCALE 3D CMOS IC TECHNOLOGIES 95
5.1. Previous Related Works 96
5.2. Impact of Interconnect Scaling on 3D Design Quality 98
5.2.1. Wire Resistance and Capacitance 98
5.2.2. Impact on Wire RC Delay 101
5.3. Impact of Technology Scaling on TSV Capacitance 101
5.4. Experimental Results 104
5.4.1. Impact of TSV Downscaling and Technology Scaling 104
5.4.2. Impact of Technology on Buffer count, Total delay and Power 105
5.4.3. Comparison of power-delay product 107
5.4.4. Influence of TSV Material Resistivity on Buffers 108
5.5. Chapter Conclusions 111

6. TSV REDUNDANCY-AWARE 3D FLOORPLANNING 113
6.1. Previous Related Works 113

6.2. Problem Formulation 115
6.3. Method of Estimation of Minimum Spare TSVs 116
6.4. Implementation of TSV Redundancy Scheme 122
6.5. Experimental Results 124
6.5.1. Impact on TSV Usage 124
6.5.2. Potential Impact on TSV Capacitance 125
6.5.3. Impact on Delay and Power Evaluation 127
6.5.4. Runtime Comparison 130
6.6. Chapter Conclusions 130

7. CONCLUSIONS AND FUTURE WORK 132
7.1. Thesis Conclusions 133
7.2. Future Scope of Work 136

REFERENCES 137

List of Tables

Table 1: Comparison of floorplan representations w.r.t their space and time complexity for 'n' modules 23
Table 2: Specifications of GSRC benchmark circuits used for our experiments 29
Table 3: Cost functions used in our current 3D floorplanner. The associated tuning parameters are taken from experimental results in [47] 32
Table 4: List of notations used in this chapter 38
Table 5: Probability of Occurrence assigned to each clustering move. The assigned value depends on stage of floorplanning and evaluated effective area ratio of floorplan 51
Table 6: Technology and TSV specifications used for experiments in this chapter 59
Table 7: Variation in floorplan packing area through iterations (n100) 61
Table 8: Comparison of wirelength and number of TSVs with Lin et al [52] using wirelength-aware cost function CF1. Number of device layers = 4. TSV D = 3 µm, H = 30 µm, p = 6 µm 66
Table 9: Comparison of proposed floorplanning (3D-DTC) with fixed-sized TSV islands approach [47] using delay-aware cost function – CF2 ((α=1.0 , γ=100) 67
Table 10: List of notations used in proposed methodology 76
Table 11: Technology Specifications used in our experiments 85
Table 12: Modified GSRC benchmarks as used in our approach. 86
Table 13: Estimated BIL using proposed approach compared with BIL in [77] for four buffer sizes. Metal Spec : 45 nm TSV Dia = 3µm, TSV Aspect Ratio = 10 88
Table 14: Comparison of number of buffers, interconnect delay and total power dissipation with [77] and [78]. TSV Specifications: Dia = 3 µm, TSV Pitch = 6 µm, KOZ = 4.6 µm. S = BUFx_8 88
Table 15: Comparison of buffer count, total delay and total dynamic power for two different TSV sizes. Average of results for all benchmarks is normalized to TSV diameter 3 µ. 92
Table 16: Metal layer (M4-M6) specifications and technology parameters for three different process nodes followed in our experiments. 102
Table 17: TSV Specifications for each size considered in our experiments 102
Table 18: Comparison of interconnect delay and power for three nodes. TSV D = 3µ, H = 30 µ, Pitch = 6 µ and KOZ = 4.6 µ 105
Table 19: Comparison of interconnect delay and power for three nodes. TSV D = 1.5µ, H = 15 µ, Pitch = 3 µ and KOZ = 2.5 µ 105
Table 20: Comparison of interconnect delay and power for three nodes. TSV D = 0.5µ, H = 5 µ, Pitch = 1µ and KOZ = 1.0 µ 105
Table 21: Electrical and thermal properties of TSV conducting materials 109
Table 22: List of minimum number of required spare TSVs for given cluster capacity 122
Table 23: Comparison of number of unused TSVs between 3D floorplanning without redundancy and 3D floorplanning with redundancy. Results evaluated on final floorplan.

No. of device layers = 4, Cost function = CF2, TSV Specifications: D = 3μm, H = 30 μm, TSV Pitch = 6 μm. 125

Table 24.Comparison of floorplanning metrics: area and delay between two cases- with redundancy and without redundancy scheme. No. of device layers = 4, Cost function = CF2, TSV Specifications: D = 3μm, H = 30 μm, TSV Pitch = 6 μm. 129

List of Figures

Figure 1: 3D ICs offer dual potential of device downscaling combined with functional diversity.... 1
Figure 2: Illustration of structure of a 3D IC with one TSV connecting two dies.... 2
Figure 3: Variation of interconnect delay and gate delay with respect to technology nodes. The sheer dominance of interconnect delay and its degrading impact on chip performance has been the driving force behind the rapid emergence of 3D ICs in the past few years.... 3
Figure 4:(a) Traditional design hierarchy (b) Floorplanning used in early layout design exploration incorporated in design hierarchy 5
Figure 5: Detailed structure of a TSV through Si substrate.... 11
Figure 6: Conceptual views of three types of 3D IC technologies : (a) Face-to-Face Bonding (b) Face-to-Back Bonding (c) Back-to-Back Bonding¬.... 11
Figure 7: Sequence pair (SP) representation. POS = positive sequence; NEG = negative sequence.... 24
Figure 8: General flow diagram of evolutionary algorithm-based 3D floorplanning as used in this work.... 29
Figure 9: Comparison of area and capacitance overhead for two different TSV placement approaches:(A)a single TSV (B) TSV island.... 36
Figure 10: Elmore-model based RC delay estimation in 3D interconnects. A two-pin 3D net containing one TSV with corresponding distributed equivalent RC circuit is shown..... 39
Figure 11: Components of TSV-to-wire capacitance (CTW). Surface and fringe capacitances between top of TSV and sidewall of TSV with overhead wires are considered as per [55]..... 41
Figure 12: Reference TSVs (highlighted yellow) subjected to maximum capacitive coupling from adjacent TSVs within a cluster, are shown for three different TSV cluster types : (a) when reference TSV is a middle-TSV (b) when reference TSV is an edge-TSV (b) when reference TSV is a corner-TSV..... 42
Figure 13(a): Illustration of TSV edge effects. Reference TSV (highlighted yellow) subjected to capacitive coupling from adjacent TSVs due to edge-effects. Coupling is influenced by relative positioning of the abutting cluster..... 43
Figure 13(b): Maximum coupling due to TSV edge effects occurs when adjacent cluster is aligned as shown.... 43
Figure 13(c): TSV edge-effects considering a corner TSV as reference TSV (note: only case of maximum coupling from nearby cluster is shown) 43
Figure 14: Merge of two clusters A and B. The area and capacity of merged cluster A is updated. Index of Cluster B joins the existing queue of empty clusters on the same device layer. Note: Labels - 'C', 'D', 'E', indicate the indices of empty clusters already present in the queue due to previous merge operations on the same device layer. A merge operation impacts the TSV footprint, packing area and decreases the population of clusters..... 47

Figure 15: TSV cluster division. The original cluster A with nine TSVs is updated with a new capacity and size (1x1). The remaining TSVs are allocated to an empty cluster E (dequeued on the same layer). A division operation impacts the TSV footprint, total packing area and increases the population of clusters 48
Figure 16: Illustration of recombination. Two clusters A and B recombine to form clusters with new capacities and sizes. Both TSV footprint and packing area are impacted due to recombination while total population of clusters remains the same.................... 50
Figure 17: Steps of dynamic TSV clustering algorithm 53
Figure 18: Design flow of 3D floorplanning with dynamic TSV clustering Area shaded in grey is repeated for each candidate floorplan in a population. 55
Figure 19: Steps of dynamic nets-to-TSVs assignment. The steps within shaded area are repeated for each 3D net in the input netlist 57
Figure 20: Variation of number of TSV clusters on top-three device layers with iterations. Data labels indicate percentage reduction in TSV cluster count as compared to base case – iteration #100. (For GSRC benchmark: n100).. 60
Figure 21: Dynamic variation in distribution of cluster sizes on three device layers at three stages of floorplanning. With formation of a wide range of cluster sizes, packing area improves by 56% from (a) stage I to (c) stage III. GSRC benchmark(n100) 62
Figure 22: Distribution of cluster sizes on final optimized floorplan for GSRC benchmarks: n200 (top) and n300 (bottom).. 63
Figure 23: Effective area ratio (EAR) achieved with proposed approach (3D-DTC) compared with two common floorplanning approaches. 3D-DTC shows 24% increase in area efficiency as compared to single TSVs placement and 6% more area efficiency as compared to fixed-sized TSV islands. 65
Figure 24: Comparison of area efficiency between three cases – (i) Fixed-sized TSV islands (ii) 3D-DTC with shared KOZ in TSV clusters (iii) 3D-DTC without sharing KOZ in TSV clusters. Results shown for case(ii) and (iii) are relative to base case (i)... 65
Figure 25. Runtime of our approach compared with floorplanning with fixed-sized TSV islands 69
Figure 26(A): Cluster size distribution when merge and divide are assigned equal probabilities of selection through all three stages of floorplanning (p=0.45)........... 70
Figure 26 (B): Cluster size distribution when merge and divide are assigned dynamically changing probabilities as per Table 5. (GSRC Benchmark: n100)......................... 70
Figure 27: Comparison of area occupied between a buffer cell and a single -TSV cell as considered in our approach. (left) a single buffer cell (8x) at 45nm (right) top view of a 3um-diameter TSV cell... 73
Figure 28(a): Comparison of area occupied between a buffer cell and a single -TSV cell as considered in our approach. (left) a single buffer cell (8x) at 45nm (right) top view of a 3um-diameter TSV cell.. 78
Figure 28(b): Elmore delay RC equivalent circuit of 2D segment 78
Figure 29: The estimated BIL is used to find candidate locations of buffers in a two-pin 3D net by traversing the net from sink-to-source 81

Figure 30. Effect of TSV clustering on TSV-capacitance 84
Figure 31: Comparison of buffer count on individual device layers (GSRC benchmark: n200). Data labels indicate percentage reduction w.r.t base case [78] 88
Figure 32: Variation of BIL with technology scaling. Data labels indicate BIL values normalized with respect to 45 nm node for each type of 3D net instance. (TSV dimensions: diameter = 3μm, aspect ratio =10)........ 90
Figure 33: Variation of 'k' with TSV downsizing and technology. The value 'k' represents the distance by which buffer moves to its optimal position (Fig.29). Data labels show percentage increase in 'k' w.r.t the base cases at 45 nm........ 91
Figure 34: Impact of TSV cluster size on the distance required to move driving buffer to optimal position before a TSV........ 93
Figure 35: Impact of interconnect scaling on wire capacitance components. 97
Figure 36: (a) Variation of unit length wire resistance and capacitance with technology scaling (b)Variation of wire RC delay with technology scaling. Wire specifications followed for semi-global (M4-M6) metal layers at each technology 98
Figure 37: Diagram of backend (BEOL) structure of VLSI circuit. (Source: ITRS [4]).. 99
Figure 38: Impact of process scaling on TSV capacitance for three TSV sizes: (a) Small Cluster (b) Large-sized Cluster........ 102
Figure 39.Impact of technology scaling on TSV delay contribution to wire delay........ 103
Figure 40: Variation of total buffer count with technology scaling and TSV downsizing for three GSRC benchmarks........ 106
Figure 41. Power-delay product (PDP) variation with technology scaling for two TSV sizes and three GSRC benchmark circuits........ 107
Figure 42. Variation of BIL with TSV material resistance. The impact of TSV material resistivity becomes dominant with increase in number of TSVs per net........ 110
Figure 43.Signal re-routing through s-TSVs 1 and 2 in a fault-tolerant TSV structure when f-TSVs 1 and 2 fail. (Source: [85]) 114
Figure 44. Number of possible situations of 'x' failed TSVs in small sized TSV clusters. 0≤x≤total(K_i) 117
Figure 45. Number of possible situations of 'x' failed TSVs in medium-sized TSV clusters 0 ≤ x ≤ total(Ki) 118
Figure 47(a): Probability of failing of 'x' number of TSVs for cluster sizes 2-15, as considered in our experiments 119
Figure 47(b). Probability of failing of 'x' number of TSVs for cluster sizes 18-30, as considered in our experiments. With increase in cluster size, probability of a single occurrence of failed TSV increases 120
Figure 48:TSV redundancy-aware 3D floorplanning with dynamic TSV clustering. The allocation of spare TSVs precedes the nets-to-TSVs assignment process in each iteration........ 123
Figure 49. Comparison of assigned TSVs on individual device layers between two cases: i) 3D floorplanning without redundancy and ii) 3D floorplanning with redundancy. Note: Device Layer 3 corresponds to the topmost active layer on the 3D chip 126

Figure 50. Comparison of total capacitance contributed by each cluster size with and without applied TSV redundancy. Increase in spare TSVs in a cluster helps contribute to reduction in cluster capacitance 127
Figure 51. Illustration of differences in nets-to-TSVs assignment with and without application of TSV redundancy scheme to individual clusters..................................... 128
Figure 52 Comparison of total interconnect power evaluated with and without redundancy. Due to the direct dependency of power on TSV capacitance, its reduction in TSV capacitance contribution to individual nets plays a more role in power reduction 129
Figure 53.Comparison of runtime between two floorplanning approaches................... 130

Glossary

3D IC	3D Integrated Circuit
3D-SOC	3D System on Chip
3D-WLP	3D Wafer Level Packaging
B2B	Back-to-Back
BEOL	Back end of line
BIL	Buffer Insertion Length
BIS	Buffer Insertion Scheme
CMP	Chemical Mechanical Polishing
CVD	Chemical Vapor Deposition
D2D	Die -to-die
D2W	Die-to-Wafer
DRIE	Deep Reactive-Ion Etching
EA	Evolutionary Algorithm
F2B	Face-to-Back
F2F	Face-to-Face
FEOL	Front end of line
GA	Genetic Algorithm
GSRC	Giga Scale Research Center
ITRS	International Technology Roadmap for Semiconductors
KOZ	Keep-out-Zone
MCNC	Microelectronics Center of North Carolina
SA	Simulated Annealing
SP	Sequence Pair
SPARC	Scalable Processor Architecture
STL	Standard Template Library
TSV	Through-silicon Via
W2W	Wafer-to-Wafer

List of Symbols

C_b	Capacitance of unit size (1x) buffer
C_{c1}	TSV coupling with adjacent vertical and horizontal TSVs
C_{c2}	TSV coupling with adjacent diagonal TSVs
C_{TT}	TSV-to-TSV coupling
C_{TSV}	TSV Capacitance
C_{TSV_max}	Capacitance of reference TSV in a cluster
C_{TW}	TSV-to-wire coupling
C_w	Capacitance per unit length of wire
CF	Cost function
D	TSV Diameter
D_{2D}	Delay of a 2D segment in a buffered wire
D_{3D}	Delay of a 3D segment in a buffered 3D net
EAR	Effective Area Ratio
H	Height of TSV
k	Distance between projected position and optimal position of driver buffer in a 3D segment.
L_{3D}	Total length of 3D segments in a net
L_{2D}	Total length of 2D segments in a net
x	Optimal distance of buffer w.r.t TSV position
L_{rem}	Known interval between TSV location and current buffer
TSV_{eqv}	TSV equivalent wirelength
L_{wire}	Estimated wirelength of a net
l_{max}	Uppermost device layer with terminal pin location of a 3D net.
l_{min}	Lowermost device layer with terminal pin location of a 3D net.
M	number of circuit modules in a benchmark
m	Number of TSVs in a row in a cluster
n	Number of TSVs in a column in a cluster
κ	Number of device layers
N_{buf}	Number of buffers in a net
N_{TSV}	Number of TSVs assigned to a net
N_{2D}	Number of 2D segments in a buffered 3D wire.
N_{3D}	Number of 2D segments in a buffered 3D wire.
N	Number of buffered segments in a buffered wire
p	TSV pitch
Φ	TSV cluster
R_c	TSV contact resistance
R_{TSV}	TSV dc resistance
R_w	Unit wire resistance
p_{inv}	Parasitic capacitance of inverter

1. INTRODUCTION

The path-breaking innovation of three-dimensional integrated circuits (3D ICs), provides a major breakthrough in meeting the ever-growing demand for device scaling in microelectronics industry and communication technology. The well-known trend of Moore's law is reaching a hard limit. By the year 2020, the minimum physical gate length of transistors will be close to 7 nm which is considered close to size of an atom. Apart from lithographic challenges of fabricating beyond 7 nm, the increased risk of electrons directly tunneling through few atomic layers of the SiO_2 insulator poses serious concern for further device scaling. In this scenario, 3D ICs are being actively explored for furthering the development of high computing chips with small form-factor and low-power consumption. With the emergence of 3D ICs as a new paradigm in the microelectronics industry, the traditional perspectives on the design, testing and usage of next-generation microprocessors are all set to change in the near future.

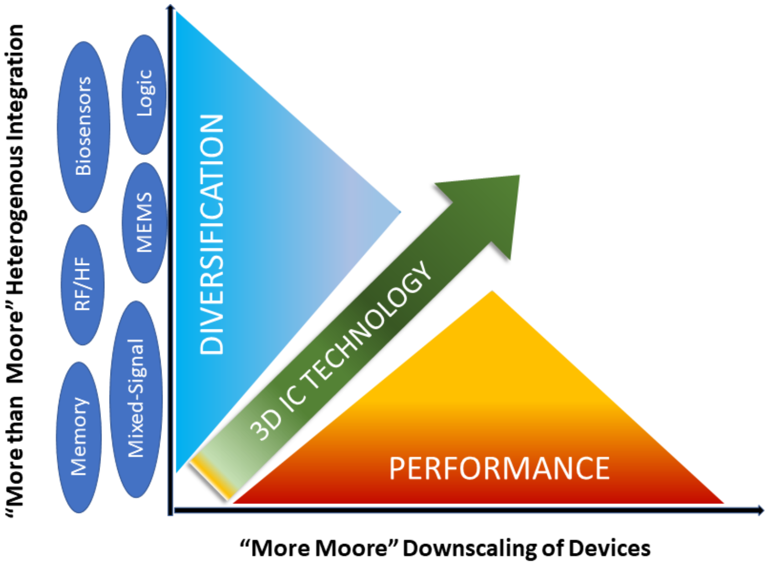


Figure 1. 3D ICs offer dual potential of device downscaling combined with functional diversity.

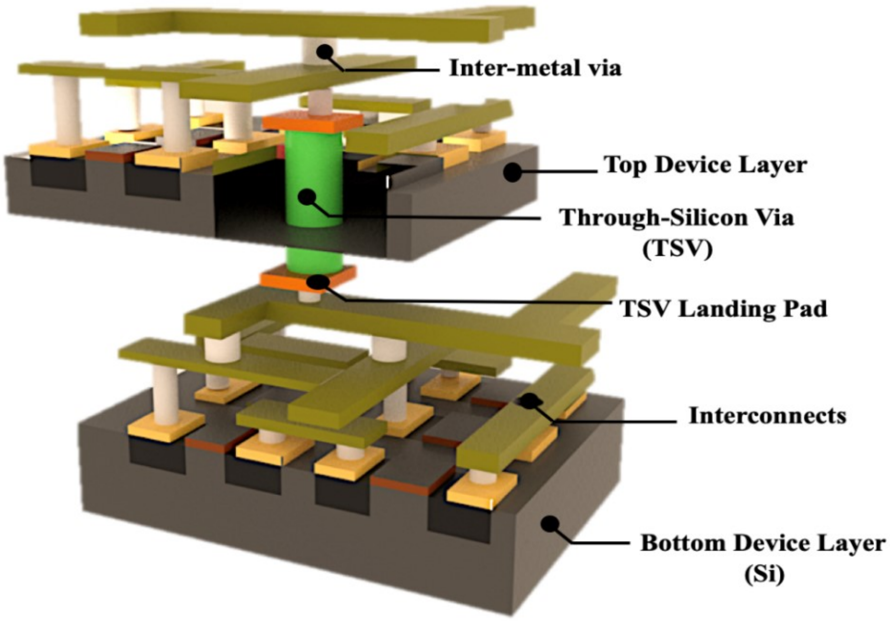


Figure 2. Illustration of structure of a 3D IC with one TSV connecting two dies.

1.1. Through-Silicon Via (TSV) based Three-dimensional Integrated Circuits

Amongst the various methods and processes for achieving 3D integration, the through-silicon via (TSV)-based 3DIC technology has emerged as the most promising and is the focus of this work. The key concept of 3D ICs is to stack individual dies and couple them with vertical interconnects i.e., TSVs, in order to achieve reduced form-factor, shorter global wires, increased device density and lower power [Fig. 2]. This promises to notably improve the performance of electronic circuits while simultaneously preserving the trend of Moore's law.

TSV-based 3D ICs differ from 3D packaging in that, 3D packaging relies on traditional methods of interconnect at the package level such as wire bonding and flip chip to achieve vertical stacks. Examples of 3D packages include *package-on-package* (PoP) where individual die are packaged, and the packages are stacked and interconnected with wire bonds or flip chip processes; and *3D wafer-level packaging* (3D WLP) that uses redistribution layers (RDL) and bumping processes to form interconnects. An example of 3D packaging is the Apple A4 SiP that places two DRAM dies on an ARM logic die and wire bonding is used for vertical interconnection.

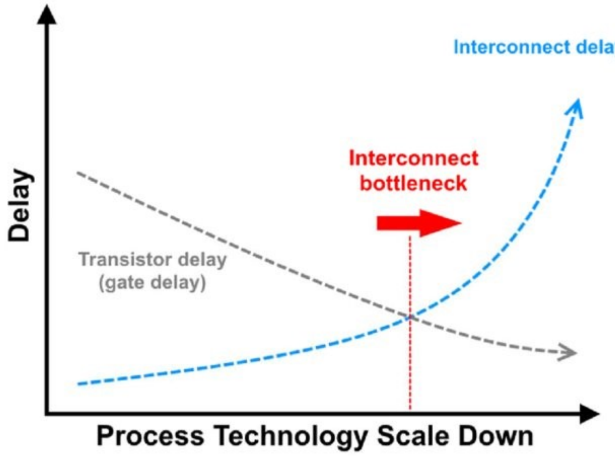


Figure 3. Variation of interconnect delay and gate delay with respect to technology nodes. The sheer dominance of interconnect delay and its degrading impact on chip performance has been the driving force behind the rapid emergence of 3D ICs in the past few years.

1.1.1 Benefits of TSV-based 3D ICs

The rapid maturation of TSV-based 3D ICs in past decade has occurred primarily due to global efforts to overcome the interconnect performance crisis in 2D ICs [2]. As shown in Fig. 3, with aggressive scaling of CMOS technology has resulted in the sheer dominance of interconnect delay over gate delay. In 3D ICs, stacking of multiple dies replaces long global interconnects with shorter vertical interconnects. The shortened global wires result in lower wire delay, therefore improving the chip performance and potentially lower the power [1][3]. In conventional 2D circuits, a large number of over-the-block wires are required for inter-block connections. Thus, more high metal layers (or global metal layers) are necessary to complete inter-block routing. On the other hand, wires in the 3D ICs are connected to TSVs, which significantly cuts down the cost of over-the-block wiring[7].

In addition to the classical downscaling of devices, a recent trend acknowledged in current and future electronic devices is the need to tightly integrate functionally and technologically diverse modules in a single chip. In this context, TSV-based 3D ICs possess the unique advantage over 2D ICs through the potential of “More than Moore” or heterogenous integration [8]. Thus, the driving force behind the widespread efforts for the commercialization of 3D ICs, is their potential to combine the gains of downscaling of devices with functional diversification in a single chip [Fig. 1].

1.1.2. Motivation

The roadmap for high-volume production of 3D ICs is still behind expectations, primarily due to the challenges involved in manufacturing, integration technologies and adequate design automation tools [6]. From the perspective of IC design, conventional electronic design automation (EDA) tools are mostly geared towards addressing the classical challenges of planar 2D ICs. The introduction of the vertical dimension in 3D ICs significantly complicates the design space for layout exploration by introducing new design constraints [7][9]. At the physical level of abstraction, there is increasing demand for developing specialized tools and methods for layout optimization, which can effectively tackle the navigation of the complex 3D solution space [10]. The most vital concern towards developing such tools is to effectively address the multitude of challenges which are unique to 3D ICs, preferably early during the layout design stage. This circumstance defines the overall motivation for this thesis work.

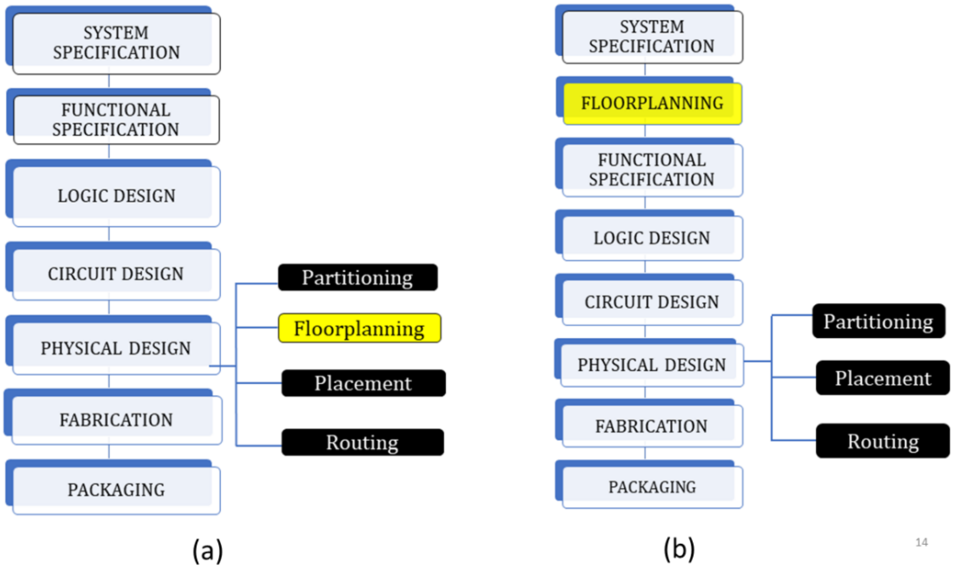


Figure 4. (a) Traditional design hierarchy (b) Floorplanning used in early layout design exploration incorporated in design hierarchy

1.1.3. Broad Overview of Thesis

On a broader perspective, this thesis focuses on developing new and efficient methodologies for early layout design exploration in 3D ICs. It is noted here that the main goal at this stage is to obtain a computationally fast and reasonably accurate evaluation of the floorplan solution quality, much prior to the placement and routing stages. Fig. 4(a) shows a traditional design hierarchy where floorplanning is performed post-partitioning at the gate-level. Fig.4(b) depicts the hierarchy which uses floorplanning at a higher level of abstraction.

The quality of a 3D floorplan is defined by performance metrics, i.e., area, wirelength, interconnect delay and dynamic power. It is critically important that these performance metrics are estimated as early and accurately as possible, as the quality of the generated floorplan sets the foundation for better chip performance at all subsequent stages of design.

Therefore, the design methods developed in this thesis are primarily focused on improving the 3D floorplan quality. Secondly, generating more realistic 3D layouts, also improves the accuracy of evaluation of the goodness of 3D floorplan. We incorporate the non-trivial design overhead of TSVs – TSV area, TSVs' parasitics and TSVs' stress effects in the developed 3D floorplanning framework. Early inclusion of TSV-related design issues at the floorplanning stage, enables better trade-off analysis and design decisions during later stages for faster convergence at lesser cost of iterations.

1.2. Thesis Contributions

The contributions in this thesis are as follows:

1. **New dynamic TSV clustering algorithm -** A new dynamic TSV clustering methodology is developed for 3D IC floorplanning, for improving the floorplan solution quality. During co-placement of TSVs with circuit blocks, TSVs are selected probabilistically forming a wide spectrum of cluster sizes on the layout. The proposed method simultaneously optimizes the size and position of TSV clusters. Also, by generating more realistic 3D layouts, the accuracy of evaluation of floorplan metrics, i.e., area, delay, wirelength and dynamic power is improved.

Key differences w.r.t approaches in related works [11][29][34][35][37][39][41]-[44] [51][52]:

i) In order to minimize TSV area, prior works have primarily relied on reducing the number of TSVs in the design. The proposed method of dynamic TSV clustering minimizes the TSV footprint, through sharing of the keep-out-zone (KOZ) around TSVs, during optimal sizing and positioning of TSV clusters on the layout.

ii) The generated floorplan is independent of any fixed-size of TSV cluster as input. This facilitates a more realistic and accurate evaluation of performance metrics i.e. area, wirelength, interconnect delay and dynamic power.

2. **Algorithm for a novel TSV-capacitance aware nets-to-TSVs assignment**- The basic idea behind the implementation of nets-to-TSVs assignment is that selection of a cluster for TSV assignment should be such that, the delay of individual 3D nets is optimized, with the following two important considerations: (i) minimize the TSV capacitance contribution to assigned net by selecting appropriate size of the TSV cluster. (ii) minimize the number of TSVs assigned to a wire based on length of the wire. The developed algorithm is an extension of the nets-to-TSVs assignment approach by Ahmed et.al in [47].

Key differences with relation to approaches in related works [31][38][47][50][62][63]:

i) The integration of nets-to-TSVs assignment with dynamic TSV clustering addresses the critical trade-off existing between the total TSV occupied area and the total capacitance contributed by TSVs. This directly enhances the accuracy of delay and/or dynamic power estimation in 3D nets during the early design stage.

ii) The solution is independent of the order of selection of nets. Also, the selection of TSV cluster for assignment is completely stochastic and non-deterministic in each iteration, thereby improving the search of the solution space.

3. **Analytical approach for improved buffer estimation in 3D interconnects** - A crucial component of early interconnect performance optimization in TSV-based 3D ICs is to develop a fast and effective method for estimation of buffers in 3D nets. To this end,

an accurate estimation of the critical distance between consecutive buffers, known as *buffer insertion length* (BIL) is required. We first develop the analytical model for optimal buffer insertion length, which is derived using a more accurate representation of the RC delay of buffered 3D interconnects. Since both TSV RC parasitics and TSV positions along the 3D net are simultaneously considered for BIL estimation, more reliable estimate of buffers, interconnect delay and dynamic power is obtained. Secondly, we develop the analytical model for estimating the optimal position of the driving buffer before a TSV in a 3D net. This model is derived using the path delay of a 3D segment (portion of 3D net consisting of a TSV, wire segments before and after TSV and buffers before and after TSV).

Key differences with relation to existing related works [76][77][78][79]:

i) The path delay of a 3D segment includes the impact of buffer parasitics, TSV RC delay and the RC delay of wire segments attached to TSV on both sides. This leads to a more accurate estimation of BIL as compared to our previous approach in [77][79].

ii) The optimal positioning of driving buffer before TSVs overcomes the need for inserting additional buffers before/ after TSVs, unlike prior methods [79]. This results in significant reduction in estimated delay and power in buffered 3D nets.

4. Implementation of TSV redundancy scheme within 3D floorplanning framework - The successful realization of 3D ICs is largely deterred by TSV failure mechanisms and TSV reliability issues**.** We develop a TSV redundancy scheme within the 3D floorplanning framework, focused on increasing the fault-tolerance of TSV clusters and improving the overall reliability of the design. Assuming a uniform TSV failure rate and an independent TSV defect distribution, the objective is to allocate the minimum required number of spare TSVs for the given size of TSV cluster, such that a cluster is fully repairable.

Key difference with relation to existing related works [83]-[86] - Our TSV redundancy scheme is implemented in the early floorplanning stage, while considering the actual impact of positioning of spare TSVs on chip area and performance. Prior works on TSV redundancy, have introduced spare TSVs on a fixed 3D layout, overlooking the impact of additional TSV area overhead on chip performance.

5. Study of impact of nanometer CMOS technology nodes and nanoscale TSVs on 3D IC design- The degree of TSVs' adverse impact on 3D design quality is heavily influence by the technology used to build the 3D ICs. Keeping this important design concern in view, we investigate the impact of three diverse nanometer CMOS technology nodes, yet to be realized for future 3D ICs - 45nm, 32nm and 22nm, on early estimation of delay, number of buffers and dynamic power. For each technology node, we consider two different TSV diameters - 3.0μm and 0.5μm, to gain more insight into the impact of nanoscale TSVs on performance and power of future 3D ICs. Additionally, this work explores five different alternative TSV fill materials with a wide range of resistivity values. Their potential effect on the critical distance between consecutive buffers in 3D interconnects is shown.

Key differences with relation to existing related works [71][72]:

i) In previous approaches, the predictions of the impact of nanotechnologies and TSV downscaling, may be limited, as they used fixed 3D layouts generated at gate-level, with positions of TSVs being fixed. None of the previous works consider the combined impact of technology scaling with nanoscale TSVs on timing optimization with buffers at the floorplanning stage.

ii) The impact of emerging new TSV fill materials on estimation of delay and power in buffered 3D interconnects has not been considered in earlier approaches.

1.3. Through-Silicon Via Process and Fabrication

The structure of a 3D IC is shown in Fig.1. In 3D ICs, TSVs are the key enabling technology, serving as vertical channels of communication between multiple vertically-stacked dies. TSVs can be used for transferring logic , power or clock signals between adjacent dies [7] . The detailed structure of a TSV is shown in Fig.5. One side of a TSV is connected to the M_1 layer (or M_{top} metal layer depending on the TSV processing technology) on the same die and the other side is connected to a micro-bump and/or a top metal layer (M_{top}) of the bottom die through a landing pad. According to the ITRS [4], the typical diameter a TSV ranges between 2µm – 10 µm and the aspect ratio, of its height to diameter, ranges between 10:1 to 20:1.

The TSV consists of a conducting material (Cu, W or other composite materials) which is electrically insulated from the surrounding silicon substrate and other TSVs, using a thin layer of silicon dioxide (SiO_2) called the *TSV liner.* The thickness of the liner also determines the TSV capacitance [54]. The presence of an additional barrier layer (e.g. TiN, TaN) prevents the diffusion of the conducting filling material into the Si substrate. Currently, the processing of TSVs is well established for high-volume manufacturing [66]. The steps of TSV process are: (i) via-formation (ii) wafer-thinning (iii) bonding and alignment. For via formation, the first step is etching using dry reactive insulated etching

(DRIE). This is followed by via-filling. TSVs are usually filled with conductive materials like copper (Cu), polysilicon (poly Si) or tungsten (W) [70]. After deposition of liner and barrier layers, wafer/die thinning is achieved using CMP, wet and plasma-etching. Thereafter, wafer/die-bonding is performed using techniques such as – adhesive bonding, metal-to-metal bonding [65].

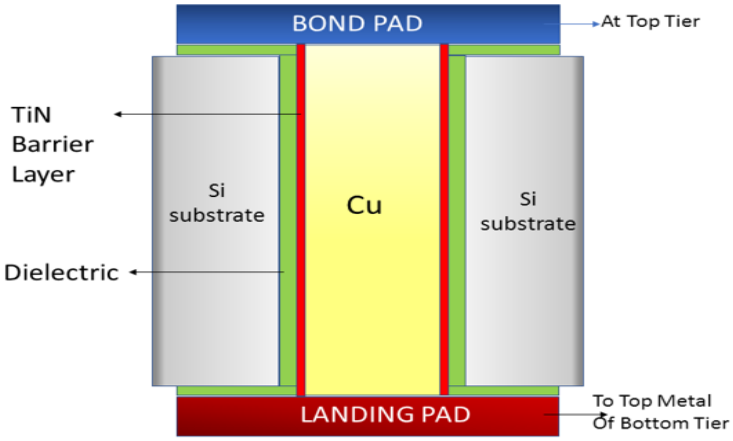


Figure 5. Detailed structure of a TSV through Si substrate

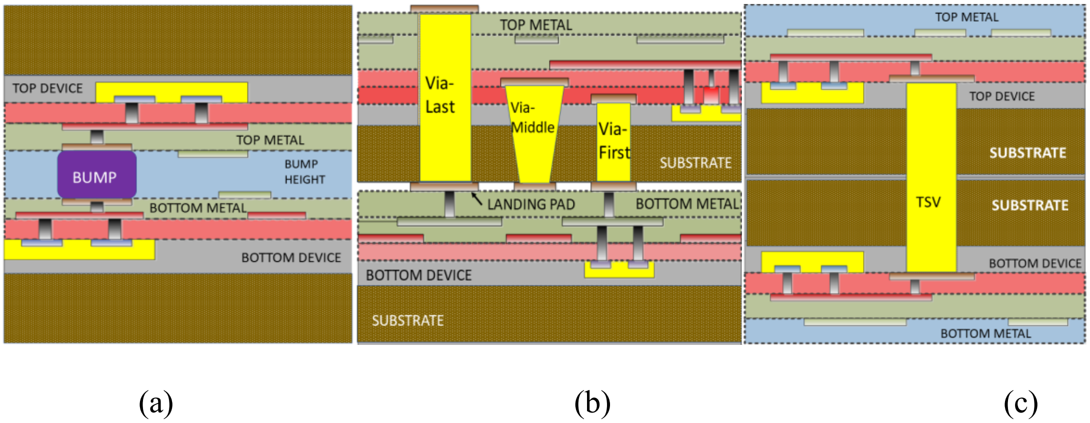


Figure 6. Conceptual views of three types of 3D IC technologies : (a) Face-to-Face Bonding (b) Face-to-Back Bonding (c) Back-to-Back Bonding

1.4. Classification of 3D ICs

Depending on the manufacturing process, the 3D IC technologies can be classified into three bonding technologies:

1. *Face-to-Face Bonding (F2F)* – In this bonding technology, vertical interconnection between dies is realized by bond-pads above the device and metal layers. Due to absence of TSVs in this technology, the possibility of TSV failure is eliminated. Therefore, the yield in F2F bonding outperforms other technologies. The die-bonding process is more evolved, as it is commonly employed in modern IC manufacturing industry. F2F bonding exhibits the highest density of interconnects, since the top metal layers directly interface with each other. However, this bonding style is not common as it allows only two dies to be integrated [Fig. 6(a)].
2. *Face-to-back (F2B) Bonding* – The F2B bonding is the most commonly used bonding technology, as it allows multilayer stacking [55]. TSVs are required to realize the vertical interconnections in this integration technology. Depending on the order of TSV fabrication in relation to the device fabrication and metallization process, the F2B integration technology can be further classified into three TSV processing types as discussed below [Fig. 6(b)]:

- **Via-first** – TSVs are fabricated before the devices (FEOL) and metallization (BEOL). The TSVs pass through the device and the substrate layers, and do not interfere with metal layers. Typical TSV diameter ranges from 1-10 μm, aspect ratio (height/diameter) – 3:1 to 10:1. Since TSVs are fabricated first, it has to withstand high-temperature device processes. The downside of this integration approach is that, the device processes may cause contamination, which may degrade the electrical behavior of TSVs [69].

- **Via-middle –** TSVs are fabricated after device processes (FEOL) and before metallization (BEOL). TSVs pass through device and substrate layers, which causes them to act as placement obstacles. Typical diameter ranges from 1-10 µm. aspect ratio (height/diameter) – 3:1 to 10:1. As compared to via-first approach, the via-middle suffers from in less contamination problems. However, the TSV processes induce substantial thermo-mechanical stress which can potentially impact the electrical behavior of the surrounding devices. To alleviate this issue, low-temperature TSV processes are typically followed in this approach.
- **Via-last** – TSVs are fabricated after device processes (FEOL) and metallization (BEOL). Since TSVs pass through metal, device layer and substrate, they cause both placement and routing obstacles. TSVs are typically large sized as compared to via-first or via-middle, due to laser drilling approach followed in this process [59]. (diameter ranges from 10-50 um, aspect ratio – up to 15:1). The large size of TSV coupled with the thermal effect of laser beam exacerbates the thermal and mechanical stress problems which may even affect or damage the components on the device layer.

The choice of TSV fabrication scheme is based on the final application requirement in the semiconductor industry. TSV technology has been developed for many applications, such as MEMS, mobile phone, CMOS image sensor (CIS), bioapplication devices, and memory products [74].

3. *Back-to-Back Bonding (B2B)* – The B2B bonding technology allows multilayer stacking. However, this technology is less common as two TSVs will be required to link

adjacent dies. This leads to decreased yield due to TSV alignment issues and also incurs a larger delay due to the TSV capacitance overhead [Fig. 6(c)]

Within the scope of this work, we use Cu-based *via-middl*e TSVs with a *face-to-back* die-bonding technique. Cu-TSVs exhibit superior electrical performance as compared to highly resistive tungsten (W)-TSVs. Also, the process of Cu-based via-middle TSVs is more established as compared to other technologies. A cylindrical TSV geometry is considered, as it is more realistic and simpler to model for electrical characteristics than other TSV types [61]. Compared to a regular square-shape via, a cylindrical via shape allows for more uniform deposition of the insulation layer which facilitates a higher breakdown voltage[60].

1.5. Challenges in TSV-based 3D ICs

Several challenging issues pertaining to the usage of TSVs and stacking of dies in the third dimension arise in 3D ICs. The maturation and success of 3D ICs eventually depends on how early and accurately these TSV-induced challenges are addressed during the layout design process:

1. TSV Footprint- TSVs occupy significant silicon area compared to surrounding devices [9]. If we consider a typical TSV of 5 μm diameter with a 2.5 μm keep-out-zone (KOZ) to ensure its mechanical reliability, then the total TSV footprint is 100um^2 which is about 5 standard cell rows in 45 nm technology [4]. Therefore, increasing the number of TSVs beyond a certain point, may counter the very benefit of wirelength reduction that one might expect, when transitioning from 2D to 3D ICs. Therefore, the number of TSVs that can be used in a design is strongly limited by the size of TSVs. As TSV footprint is determined by both the size and the number of TSVs, finding the optimal number of TSVs has become critical during early layout design stage.

2. TSV placement topology – The choice of the TSV distribution and physical positioning of TSVs on the layout relative to each other, has a decisive impact on the design quality. For e.g. random placement of single TSVs on the layout may decrease the wirelength, however with increase in size and complexity of circuit, such a topology might outweigh the wirelength benefit due to large consumption of silicon area [45]. Studies have also shown that regular uniform placement of TSVs might be beneficial to minimize TSVs' stress influences, but these schemes cause significant blockages during routing and buffer allocation [40]. Apart from stress effects, TSVs' positions also impacts the signal transmission delay in 3D nets, due to capacitive coupling effects with other neighboring TSVs and wires. Hence, the key to successful design of TSV-based 3DICs is not just in optimizing the number of TSVs, but also lies in finding the optimal position of TSVs on the layout.

3. Thermal and Mechanical Issues – Large sized TSVs induce significant thermo-mechanical stress in their surrounding silicon [82]. This stress is primarily generated due to mismatched coefficients of thermal expansion (CTE) of filling materials for TSV such as Cu or W and the surrounding substrate (Si). Therefore, it is required to avoid placing devices in this high stress area surrounding a TSV, by incorporating a safety-zone called Keep-Out Zone (KOZ). With increase in the number of TSVs in the design, the required KOZ also increases due to cumulative stress interaction between adjacent TSVs [81].

4. TSV RC Parasitics - TSVs have non-negligible parasitic resistance and capacitance (RC) which significantly impacts the delay and power dissipation of a 3D net [46][47]. The degree of impact of TSV parasitics on net delay is also dependent on the technology node [71]. The TSV RC characteristics is different from wire RC characteristics. Therefore, the use of wire RC models for TSVs could lead to significant underestimation in delay and power estimation of 3D interconnects [60]. Early and accurate characterization of TSV capacitance is an important challenge towards determining the timing performance, noise and power consumption in the circuit[57]. TSV capacitance is strongly dependent on the TSV width and height, liner-oxide thickness. It should be noted that at higher operating clock frequencies (> 3GHz), the effects of TSV self-inductance also needs to be considered [54].

5. TSV Assignment to Nets - The assignment of TSVs to individual 3D nets (nets that have pins on multiple dies) constitutes a critical problem in 3D ICs, as it determines the number of TSVs used per 3D net as well as the position of each TSV along the net [62]. Both the number and position of TSVs along the wire, influence buffer requirements [77]. The position of TSV affects the total length of the wire, therefore impacting wire delay and wire power dissipation [63]. Another key aspect of TSV assignment is that the number of TSVs used in individual nets, is critical, especially in short wires, due to the dominance of TSV capacitance over wire capacitance [61][62][63]. To account for the above impacting factors, the nets-to-TSVs assignment problem needs to be a part of floorplan optimization process, in contrast to prior approaches where TSV assignment was performed post-floorplanning stage on a fixed layout.

6. TSV Reliability and Yield of 3D ICs - The key obstacles to widespread adoption of TSV based 3D ICs is concerns about TSV reliability and yield. The yield of TSVs based 3D-ICs is limited under current manufacturing process. Only one defective TSV can fail the entire chip with all known-good dies [85]. Post floorplanning stage placement of TSVs heavily, relies on the efficient distribution of whitespace and does not consider the allocation of appropriate KOZ to ensure the mechanical reliability of TSVs and surrounding gates due to stress. Low mechanical reliability can severely degrade chip performance. Hence, an optimal TSV planning should allocate TSVs considering both reliability and chip yield.

1.6. Thesis Organization

The rest of this dissertation is organized as follows:

- In Chapter 2, the 3D floorplanning problem is formulated. The floorplan representation for non-slicing floorplan as used in the scope of this work is discussed. The different stages of evolutionary-algorithm (EA)-based 3D floorplanning followed in this work are enumerated and described.
- In Chapter 3, the methodology of dynamic TSV clustering is discussed. The mutation operators for executing the TSV clustering moves are described. The conditions for probabilistic selection of TSV clustering moves during 3D floorplanning are discussed. The steps of algorithm for dynamic TSV clustering are described. Experimental results with TSV clustering are presented using two types of cost functions – (i) delay-aware cost function and (ii) wirelength-aware cost function.
- Chapter 4 describes the novel TSV cluster capacitance -aware nets-to-TSVs assignment procedure incorporated in our 3D floorplanning framework.

- Chapter 5 focuses on a novel scheme of TSV-position aware buffer insertion incorporated in our 3D floorplanning framework. The developed approach stresses on the utmost importance of accurately characterizing the delay of a 3D segment in a buffered wire. This leads to improvement in the buffer estimation in 3D wires, consequently improving the delay and power evaluation of 3D wires.
- Chapter 6 describes the newly built TSV redundancy scheme within our 3D floorplanning with TSV clustering approach. The methodology of allocation of redundant TSVs to individual clusters during cluster sizing process is discussed. The impact of 3D floorplanning with redundant TSV allocation on chip performance is discussed.
- The final conclusions and scope of future work are presented in Chapter 7.

2. STATE-OF-THE-ART IN 3D IC FLOORPLANNING

Floorplanning constitutes the first major step in physical design, which determines the placement of the major blocks and/or macros and other components on the layout [11]. The significance of this design step lies in the fact that the resulting floorplan influences the optimization of all the subsequent stages i.e. placement and routing. The decisions taken for macro/block placement, partitioning, I/O-pad placement, and power planning directly or indirectly impact the overall implementation cycle [12]. Floorplanning also provides early feedback that evaluates architectural decisions, estimates chip area, delay and congestion caused by wiring. With aggressive scaling of technology every couple of years, both the size and complexity of circuit is increasing rapidly. As a result of the increasing design complexity, the usage of hierarchical design and intellectual property (IP) modules has inevitably grown in recent years [42]. This trend makes floorplanning much more critical to the quality of a very large-scale integration (VLSI) design than ever[26].

2.1. Need for Early 3D Layout Design Exploration

The adoption of TSV-based 3D integration as a viable commercial technology, is hindered by lack of a standardized 3D IC design methodology. Existing tools for 2D floorplanning lack the capability for effective exploration of the 3D solution space. To harness the full potential of 3-D ICs, effective tools for early layout design exploration are required, which enable accurate characterization of the physical and electrical impact of TSVs. Such tools can equip the designers with an early knowledge of obtainable trade-offs between the optimized parameters, whether the design is satisfying the required layout constraints, therefore speeding up convergence.

3D EDA tools for early layout exploration can also provide designers a powerful means to get a head-start on the placement and budgeting the routing resources later in the design cycle. Prior studies have shown that ignoring TSV impact during early layout planning can lead to significant deviation in area and wire estimation [43][44]. This will result in erroneous design choices, failure to achieve convergence and increased cost due to design iterations. Hence, the demand for efficient TSV-aware floorplanning tools that can be used to build high performance digital systems as well as shorten design time.

2.2. 3D Floorplanning Problem Formulation

Let B = $\{b_1, b_2..., b_m\}$ be a set of '*m'* rectangular blocks with given aspect ratio ($h_1/w_1, h_2/w_2......, h_m/w_m$). N = $\{n_1, n_2...., n_m\}$ represents the list of nets connecting pins located at the center of each block. Let (x_i, y_i) denote the coordinate of the bottom-left corner of each block. A floorplan F is an assignment of (x_i, y_i) for each block on 2D layout such that there are no overlaps of the blocks. The objective of classical 2D floorplanning problem is to optimize the area of the chip (i.e. minimum axis-aligned rectangle that contains all floorplan blocks) and total inter-module wirelength under fixed-outline constraints.

The optimization of chip area is not just dependent on finding the optimal (x,y) positions of the blocks, but is also dependent on the optimizing the shape (aspect ratio) of the individual modules. From the perspective of manufacturing and packaging requirements, an important consideration during chip area optimization is to keep the chip aspect ratio as close as possible to a desired value. For instance, a square chip (aspect ratio ≈ 1) may be preferable to a non-square chip. It is therefore evident that the chip area and aspect ratio of chip are inter-related, and hence, should be considered together.

The input set of blocks can consist of both *hard* and *soft* modules. Hard blocks are relevant in case of floorplanning with pre-existing blocks or intellectual property (IP). The aspect ratio and areas of hard blocks are fixed, whereas, for a soft block, the area is fixed but the aspect ratio is changeable. In our floorplanning problem we only consider hard macro/blocks that have fixed dimension (width and height). However, in case of *soft modules* the aim of floorplanning also includes finding a desired aspect ratio for each soft module, while optimizing the total chip area.

Problem Formulation for 3D IC Floorplanning - The 3D IC floorplanning problem is to generate a floorplan F, which is an assignment of (x_i, y_i, z_i) for each b_i, $1 \leq i \leq m$ on 'κ' device layers, such that $1 \leq z \leq$ 'κ' such that there is no module overlap within the same device layer. The goal of 3D floorplanning is to optimize a predefined cost function such as a linear weighted combination of chip area (maximum chip footprint multiplied with number of device layers), wirelength which is the sum of interconnect lengths and number of TSVs (number of vertical inter-layer connections), total interconnect delay or power dissipation.

2.3. Classification and Representation

2.3.1. Floorplan Structures

Floorplans are classified into two types based on layout structures - *slicing* and *non-slicing* floorplan[15]. A slicing floorplan is attained by repetitively slicing the floorplan region horizontally or vertically. A slicing floorplan is represented using a binary tree, known as slicing tree, with modules at the leaves and type of cut at the internal nodes. The two types of cuts are H (horizontal), for dividing the floorplan left or right, and V (vertical), for dividing the floorplan in top or bottom region. The non-slicing floorplan is represented using horizontal constraint graph (HCG) and vertical constraint graph (VCG) and defines the horizontal and vertical relationship between blocks or modules [13].

2.3.2. Floorplan Representations

The classical floorplan optimization problem is a complex NP-hard problem, which requires the use of meta-heuristic search algorithms to arrive at a near-optimal solution in polynomial time. However, to apply any search technique, a floorplan needs to be first encoded as a candidate solution termed as *floorplan representation*. The floorplan representation is an abstract representation, which defines the geometric relationship of modules in a floorplan [15]. In the context of 3D ICs, the topological relationship between modules (i.e. circuit blocks, gates or TSVs) needs to be represented for individual device layers. The choice of floorplan representation becomes an important determinant of performance of floorplanning algorithms, as it determines the complexity of the 3D solution space and the computational cost of searching or traversing the solution space [32]. Two important characteristics of an efficient floorplan representation are: i) the representation must result in a unique floorplan topology and ii) transforming a representation to floorplan should take least amount of time.

Sliceable floorplans are modeled using binary trees that can be represented using a *Normalized Polish Expression*. Non-slicing floorplans are modeled using *constraint graphs* in which each node denotes a module and each edge represents a topological relationship between modules. The graph modeling the horizontal topological relationship (left-to-right) between modules is termed as the *horizontal constraint graph (HCG)* whereas the top-to-bottom relationship between modules is represented by the *vertical* *constraint graph (VCG)* . Several representations have been proposed for non-slicing and general floorplans such as - *Bounded Slicing Grid Structure (BSG)* [17] [18], *Corner Block List (CBL*) [14], *Corner Sequence (CS)* [16], *Sequence Pair (SP)* [15] [19] [26], $B*$ *Tree* [18] [20], *Transitive Closure Graph (TCG)* [21] [22],*O- Tree* [25].

2.3.2.1. Floorplan Representations for 3D ICs

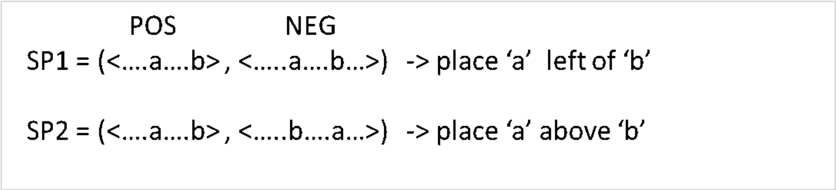
In the context of 3D ICs, floorplan representations are classified as: (i) true-3D and (ii) quasi-3D representations [35]. A true-3D representation is an extension of 2D representation to a 3D structure e.g. 3D slicing tree and sequence triple. However, a true-3D representation incurs a big space and time penalty due to increased redundancies in the z-axis data structure. A quasi-3D representation is basically an array of 2D representations, each 2D representations belonging to an individual device layer e.g. two-layer BSG, four-layer TCG.

Table 1. Comparison of floorplan representations w.r.t their space and time complexity for 'n' modules

Representation	**Floorplan Type**	**Solution Space**	**Time complexity**
Binary Tree	Slicing	$O(n!\,2^{5n-3}/n^{1.5})$	$O(n)$
Normalized Polish Expression	Slicing	$O(n!\,2^{3n}/n^{1.5})$	$O(n)$
Bounded Slicing Grid (BSG)	Non-slicing	$n!\,C(n^2,n)$	$O(n^2)$
B* Tree	Compacted	$O(n!\,2^{2n}/n^{1.5})$	$O(n)$
Corner Block List (CBL)	Mosaic	$O(n!\,2^{3n})$	$O(n)$
O-Tree	Compacted	$O(n!\,2^{2n}/n^{1.5})$	$O(n)$
Transitive Closure Graph (TCG)	General	$O((n!)^2)$	$O(n^2)$
Corner Sequence (CS)	Compacted	$\leq (n!)^2$	$O(n)$
Twin Binary Sequence	Mosaic	$O(n!\,2^{3n}/n^{1.5})$	$O(n)$
Sequence Pair (SP)	General	$O((n!)^2)$	$O(n^2)$

2.3.2.2. Sequence Pair (SP) Representation for 3D Floorplan

The sequence pair or SP is a flexible representation for modeling general floorplans and was first proposed by Murata et al. in [15] for rectangular module placement. An SP representation consists of an ordered pair of sequences of module indices - positive sequence and negative sequence. The sequence of occurrence of module indices in both the sequences determines the geometric relation of the modules as per the rules shown in Fig. 7. Using the SP, the modules can be placed on a grid structure, and corresponding constraint graphs can be constructed to evaluate cost. For example, (124536, 326145) can represent a floorplan of the six modules 1, 2… 6.



```
           POS           NEG
SP1 = (<….a….b>, <…..a….b…>)  -> place ‘a’  left of ‘b’

SP2 = (<….a….b>, <…..b….a…>)  -> place ‘a’ above ‘b’
```

Figure 7. Sequence pair (SP) representation. POS = positive sequence; NEG = negative sequence

2.4. Existing Approaches for 3D Floorplanning

The classical approaches for 2D floorplanning is well studied. In the realm of 3D IC floorplanning, researchers have attempted to adapt the existing 2D floorplanning algorithms for the exploration of the vertical dimension [28][33][[34][35]-[42]. The primary challenge in these approaches is the inclusion and modeling of TSVs in the floorplanning problem. In general, approaches to floorplanning belong to the following classes – i) constructive ii) iterative and iii) knowledge-based. With constructive approaches, an initial seed module is selected, and modules are added one-by-one to the partial floorplan. Iterative algorithms start from an initial floorplan represented by an abstraction and iteratively perturb the solution. The process continues until the termination or until a feasible optimal floorplan is obtained with no further improvement in cost of floorplan. The knowledge-based algorithms rely on a knowledge base containing data about the floorplanning problem, set of rules describing the manipulation of data to converge to a near-optimal solution and an application for applying the rules to the knowledge-base. [12]

The continued trend of scaling of devices and increasing circuit complexity has resulted in the widespread use of intellectual property or IP modules and high-level functional units in large-scale designs[42]. To cope with the layout planning challenges in presence of IP modules, certain classical 2D floorplanning approaches have now become popular for 3D IC design, such as – i) analytical approaches and ii) simulated annealing. TSV-aware analytical placement approaches have been proposed in [29][30]. The analytical 3D placement approaches such as integer-linear programming (ILP) [31], apply mathematical programming consisting of an objective function (cost metric i.e. area and wirelength) for floorplan optimization and a set of constraints (i.e. non-overlapping and aspect ratio restrictions).

The simulated annealing (SA) algorithm is an iterative, non-deterministic heuristic technique widely applied floorplanning[11][12][13][19]. To apply SA, a suitable floorplan representation, which encodes the geometric proximity of modules is required. Given an initial floorplan ‘S’ and an initial temperature ‘T’, the SA provides a non-zero probability to move from the current solution *S* to a neighboring one ‘*S*”, even with a higher cost of move. With this uphill move capability, a globally optimal solution can be reached irrespective of the initial solution. The probability of accepting an uphill move depends on – (i) the magnitude of the move (cost(S') – cost(S)) and (ii) the search time (rate of annealing of temperature 'T'). Unlike the analytical methods, the advantage of SA approach is that all intermediate solutions are feasible. In a nutshell, the SA relies on the representation of the geometric relationship among modules, whereas in analytical approach the absolute topological relationship between modules is captured directly.

More recently, the iterative approach of floorplanning has further evolved with the application of nature-inspired population-based metaheuristic optimization techniques such as - evolutionary algorithm (EA)[33][35][47]and genetic algorithm (GA) [23][24]. Both methods are based on non-deterministic and stochastic search applied on a set of candidate solutions. The basic premise of EA is based on principles of biological evolution and consists of four mechanisms – (i) initialization of parent population (ii) reproduction or mutation (iii) selection (iv) termination. The steps provide an easy and efficient way to modularize the implementation of EA-based algorithms.

2.5. Methodology of Data Generation

The work presented in this thesis is built on the initial version of the 3D floorplanning software developed by R.K. Nain [35][87]. This version of software is based on evolutionary algorithm (EA) and uses sequence pair representation. This version of the tool placed circuit blocks across multiple devices while vertical constraints were applied on the location of sub-modules. However, the physical area and position of TSVs were ignored while estimation of wirelength. As a result, the final area and wirelength were underestimated in this version. The initial procedure for nets-to-TSVs assignment was built by M.A. Ahmed [47] in the second version of the 3D floorplanning software. This built-in nets-to-TSVs assignment enabled simultaneous consideration of the TSV physical area and TSVs' positions along the assigned net for improved estimation of wirelength through subnet method [62][63]. For net-delay estimation, the resistance and capacitance models for TSVs were also included by M. A. Ahmed [55][61]. Unlike the previous version which optimized wirelength and number of TSVs separately, the second version of the 3D floorplanning tool facilitated direct optimization of total interconnect delay. However, the generated floorplan solution was limited due to the assumption of fixed number and size of TSV islands irrespective of input benchmarks. The possibility of directly optimizing TSV area for performance improvement was also ignored.

The detailed flow and method of data generation in the current version of the 3D floorplanning tool is provided in the following sections.

2.5.1. General Assumptions of 3D Floorplanning

Based on the technology requirements and recommendations of the International Roadmap of Semiconductors (ITRS) [4], the current version of the 3D floorplanning tool has been developed in this work with the following assumptions:

1. The device layers are stacked using face-to-back (F2B) die-stacking strategy, because it is the most commonly used configuration, and also doesn't limit the number of device layers that can be stacked.
2. Only signal TSVs are assumed for our experiments. The impact of power-ground TSVs and thermal TSVs is beyond the scope of the thesis.

3. Cu-based via-middle TSVs are used because of their established process and superior performance.
4. The maximum number of stacked device layers assumed in our experiments is four. The stacking of more than four device layers may counter the benefit of wire reduction due to the increased silicon surface area of TSVs in the design [45].
5. During assignment of nets to TSVs, it is assumed that a 3D net spanning between two consecutive layers needs allocation of a single TSV on the upper layer only.
6. A common requirement in modern ASIC designs is fixed-outline of chip. Keeping this in view, a fixed -outline constraint is applied during our floorplanning, i.e., all modules (circuit blocks and TSV clusters) are packed within the fixed -outline region with an aspect ratio of close to one. The circuit blocks are hard modules with fixed area and aspect ratio. A maximum allowed whitespace of 15% is assumed. Some important terminologies specific to our floorplanning tool are –

- Optimum chip area (*chip_optiarea*) – summation of area of blocks and TSV clusters.
- Fixed area *(fix_area)* – summation of area of blocks, TSV clusters and the white space area.
- Packing area *(chip_packarea*) – Represents the evaluated floorplan area. It is calculated by the *chip_w*chip_h*nlayer*, where *chip_w* and *chip_h* represents the width and height of the minimum rectangle that encloses the blocks and TSVs. The *chip_w* and *chip_h* is the maximum value of all the device layers represented as *nlayer*. The packing efficiency of a floorplan is represented as the ratio of *chip_optiarea* and *chip_pack area*.

Table 2. Specifications of GSRC benchmark circuits used for our experiments

Benchmark	Total Block Area		Total #of nets	
	Original (mm^2)	Expanded (mm^2)	Original	Expanded
N100	0.176	17.64	885	885
N200	0.187	18.67	1585	2136
N300	0.273	27.32	1892	2914

7. **Benchmark Specifications** – Table 2 shows the original and modified GSRC benchmark circuits used in our experiments. GSRC benchmarks are commonly used circuits in existing works and are ideal for experimentation, as compared to other older benchmarks (ami and MCNC). The block area expansion and modification in number of nets for each benchmark is reported. The modification of benchmarks was necessary for comparison with existing works. Keeping in mind that the current growth of VLSI circuit sizes and complexity, the benchmarks have been modified by expanding each block area by 100x and adding more multi-pin nets to the original netlists.

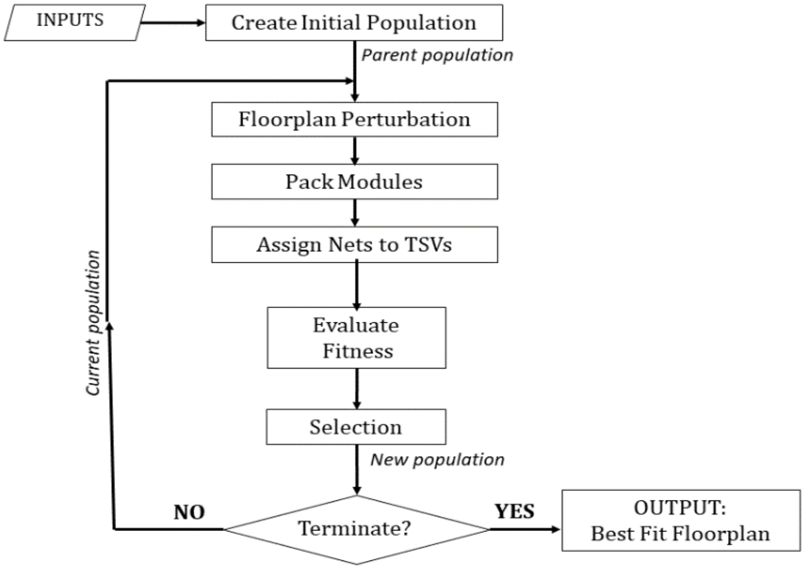


Figure 8. General flow diagram of EA-based 3D floorplanning as used in this work.

2.5.2. Evolutionary Algorithm based 3D Floorplanning Flow

This section describes the basic strategy for data generation using the developed floorplanning software. The data reported in our experimental results in subsequent chapters are obtained on the final best fit floorplan. The objectives to be optimized depend on the cost function used during floorplanning. Fig.8 shows the flow diagram of evolutionary algorithm-based 3D floorplanner used to obtain the final floorplan.

1. **Floorplanning Inputs** – The floorplanning inputs include – list of blocks with specified area and aspect ratio, netlist, technology specifications for semi-global metal layers, TSV dimensions, TSV material resistivity, wire resistivity, TSV pitch, look-up table for KOZ dimensions corresponding to each cluster size, supply voltage, operating frequency and buffer specifications (buffer output resistance and input capacitance).The TSV dimensions, pitch and KOZ are required for evaluation of packing area. The metal dimensions and resistivity and buffer specifications are required for evaluation of interconnect delay, power and buffers.
2. **Creating Initial Population -** An initial set of floorplans called the *parent population* is randomly created. Unlike analytical methods or simulated annealing (SA), an evolutionary algorithm (EA) processes a population of candidate solutions in parallel. The size of the population is selected as twenty.
3. **Floorplan Perturbation** – In each iteration, each floorplan in the current population undergoes a *perturbation* to result in a *child* floorplan. A floorplan is perturbed by randomly selecting a mutation operator or *move*, from a set of predefined moves. Due to the exclusive use of mutation operators in the 3D floorplanner, a feasible parent always produces feasible offspring. Some of the moves were included in the initial

software inherited from [57] are : *(a) swap (b) invert (c)rotate (d) exchange and (e)change group*. In second version of the software, as additional *random-swapping of nets* between TSV islands was included [31]. The current developed version of the software includes the dynamic TSV clustering moves in addition to the above moves. The selection of each move is probabilistic, and the assigned probability of each move depends on the stages of floorplanning and the quality of floorplan as it proceeds towards convergence.

4. **Fitness Evaluation** – After perturbation, each floorplan is evaluated using a pre-defined cost function. The cost function is a weighted sum of optimized metrics i.e., area, delay, number of TSVs, wirelength, power. Based on evaluated cost function each floorplan is assigned a rank or fitness value. Hence, the order of floorplans in a population based on fitness value may change if the optimized objectives change. Table 3 shows the two cost functions CF1 and CF2 used for our experiments [47]. The appropriate values of associated weights for each metric in a given cost function are determined through empirical observations.
5. **Selection** - A selection process is applied to the pool of parent and child populations to select the best fit individuals to enter the next generation. In the scope of this work, the 3D floorplanner uses a subset of evolutionary algorithm known as evolution strategy (ES), in which the selection process is purely deterministic. The floorplanning run terminates if any of the following criteria is met:
    - The desired packing area is achieved.
    - Number of iterations reaches maximum specified limit.
    - Fitness value repeats for a specified number of iterations.

Table 3. Cost functions used in our current 3D floorplanner. The associated tuning parameters are taken from experimental results in [47]

Cost Functions	*Optimized Metrics*	*Notations*	*Fitness Function*	*Associated Weights*		
				β	γ	λ
CF1	Packing Area	Area	$CF1 = Area + \beta * WL + \gamma * nTSV$	0.01	100	-
	Total Wirelength	WL				
	#TSVs	nTSV				
CF2	Packing Area	Area	$CF2 = Area + \lambda * total\_delay$	-	-	50
	Total Interconnect Delay	Total_delay				

After termination of floorplanning algorithm, the best fit floorplan is selected from a set of floorplans, as the final 3D layout. Depending on the selected cost function, the chip area, total wirelength, number of TSVs, delay or dynamic power are evaluated on the final floorplan. For each estimated parameter shown in our results, we have considered the average of twenty floorplanning runs.

3. DYNAMIC THROUGH-SILICON VIA CLUSTERING IN 3D IC FLOORPLANNING

One of the most important challenges in realizing high-performance 3D circuits is early optimization of interconnect delay [54][55]. At the physical level of abstraction, interconnect delay is directly controlled by the quality of the 3D floorplan. 3D floorplanning is a complex combinatorial optimization problem, in which the quality of generated solution is fundamentally impacted by the heuristics guiding the search and evaluation of floorplan [11][12][15]. Therefore, in recent years, the demand for developing more efficient algorithms for navigating the complex 3D solution space, has justifiably grown.

3.1. Previous Works on TSV-Aware 3D IC Floorplanning

The NP-hard complexity of 3D IC floorplanning problem has spurred the development of several meta-heuristic techniques, mostly based on simulated annealing (SA)[6][24][41]. Quiring *et al.* [48] proposed a guided SA-based global-interconnect driven 3D floorplanner, to place TSVs in a fixed-outline floorplan, while considering interconnect area and routability as objectives. Hsu *et al.* [29] proposed a non-linear analytical 3D placement approach using a weighted-average wirelength model. Lin *et al.* [52] used a fast analytical-based approach in their routability-driven 3D floorplanning. Saha *et a*l. [50] proposed a multi-objective (MO) scheme to perform the TSV placement and assignment, by simultaneously optimizing four objectives- power density, wire congestion, TSV boundary distance and wirelength. Knechtel *et al.* [46], primarily focused on thermal characteristics of 3D processors. The thermal management was further incorporated into their work in [51] in which, wirelength and thermal-driven clustering of TSVs was performed. They imposed module-alignment constraints which results in greedy shifting of TSVs in case of overlap. In [51], a single capacitance value is used for all TSVs, irrespective of their placement as a TSV group or as individual TSVs thereby, limiting solution accuracy. The above studies also overlook the significant contribution of TSV-induced RC delay to individual net delay. Few works have emphasized the non-trivial impact of TSVs' RC parasitics on interconnect delay [53][61][62][78]. Kim et al.[78] evaluated the total interconnect delay, including delay contributions of TSVs and buffers, on a fixed layout and did not use it as an objective for floorplan optimization. Also, the scattering of single isolated TSVs with undefined pitch between adjacent TSVs, limits the accuracy of TSV capacitance estimation. Ahmed et al. [61] showed both peak delay and total delay reduction. However, the fixed nets-to-TSVs assignment at the beginning of the floorplanning which restricts the solution.

Prior TSV-based 3D design approaches have mostly focused on separate optimization of wirelength and number of used TSVs [35][37][38][45]. Tsai *et al*. [43], proposed a two-stage 3D floorplanning, in which the first stage used the available whitespace to place TSV blocks. In second stage, wirelength was further minimized, through deterministic reassignment of TSVs among the TSV blocks on the final floorplan. Although they achieve a compact floorplan, their solution is severely limited by the available whitespace in the final floorplan. Li *et al*. [44], proposed co-placement of TSVs with circuit blocks. However, a deterministic partitioning algorithm was used to permanently assign blocks to device layers, to minimize the number of TSVs. Although their approach reduces the solution space, many potentially superior solutions are eliminated. Force-directed methods for placement of modules and TSVs in 3D ICs were proposed in [4]. Lyu *et al.* [49] base their wirelength and TSV count minimization on force-directed partitioning algorithm. However, the position of TSVs is ignored in their partitioning, which is a major constraint.

TSVs in islands has been considered in [42][46][51][35] citing advantages from the perspective of heat dissipation, TSV redundancy and reduced effects of TSV-induced stress. However, the impact of grouping TSVs on interconnect delay optimization is not included.

3.2. Chapter Contributions

In this chapter, we develop an efficient methodology to improve the 3D floorplan solution quality. By generating more realistic 3D layouts, we seek to improve the accuracy of evaluation of the goodness of a 3D floorplan. Towards efforts to minimize interconnect performance, prior approaches have mostly focused on minimizing total wirelength and number of TSVs in a design, while the area and position of TSVs are kept fixed on the layout. Secondly, the critical dependency of TSV capacitance on the distribution and arrangement of TSVs on layout has been overlooked. In this paper, we seek to address this gap through the following contributions:

1. We develop a new dynamic TSV clustering methodology for 3D floorplanning. Our algorithm simultaneously optimizes the sizes and positions of TSV clusters, during TSVs' co-placement with circuit blocks. The generated solution is independent of any fixed input size or capacity of TSV cluster. The proposed approach facilitates more realistic and accurate evaluation of performance metrics i.e, area, total interconnect delay and interconnect power.

2. A novel nets-to-TSVs assignment method is integrated with dynamic TSV clustering-based 3D floorplanning. The delay of individual 3D nets is effectively optimized in two ways – i) by minimizing the number of TSVs assigned to a net and ii) considering the capacitance contributed by a TSV in a cluster. The optimization of net delay considers the critical trade-off between total TSV occupied area and TSV capacitance.

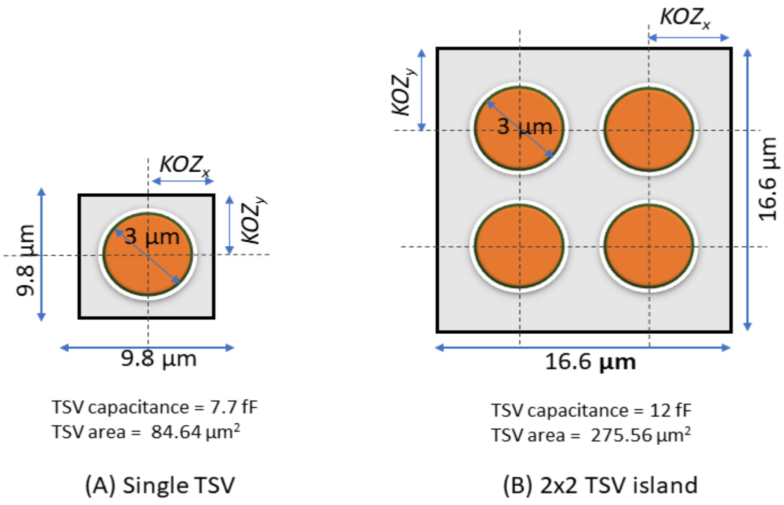


Figure 9. Comparison of area and capacitance overhead for two different TSV placement approaches: (A)a single TSV (B) TSV island.

3.3. Design Overhead of TSVs

TSVs induce significant area and delay overhead in 3D ICs [40][42][43]. The degree of TSVs' impact on a 3D design not only depends on TSVs' dimensions and material properties, but it is also influenced by the choice of technology node for building 3D ICs [72]. At 45 nm technology, the area of a 5µm-diameter TSV can be up to 10x larger than the standard cells [45]. The TSV area overhead gets further exacerbated due to inclusion of the TSV landing pad and a mandatory keep-out-zone (KOZ) [80]. The KOZ represents the area around a TSV where gate-placement is forbidden, in order to eliminate TSVs' stress influences on the mobility of charge carriers in surrounding devices [82]. The KOZ around a TSV depends on the TSV diameter, mechanical properties and thermal properties of the TSV and the surrounding silicon. Considering a cylindrical TSV of 3 µm-diameter requires a KOZ of about 4.6 µm [80], the area of a TSV cell with KOZ is about 12x larger than the TSV area without KOZ. It is evident that, excessive usage of single isolated TSVs may even negate the wirelength benefit of 3D ICs due to a substantially large increment in silicon area. Large-sized TSVs also incur significant delay overhead in the wires containing them, due to capacitive coupling with overhead wires and other TSVs in their vicinity [55][56].

Although minimizing TSV usage is essential to curb their adverse impact on chip performance, it is equally relevant to optimize the distribution and position of TSVs on the layout. To demonstrate this, we compare two different TSV arrangement schemes for a 3µm-diameter TSV – i) single TSV cell with KOZ and ii) TSV island as shown in Fig.9. The TSV island includes the appropriate pitch between adjacent TSVs and the required KOZ on the periphery. The KOZ is estimated based on analytical model in [80]. Values of TSV capacitance are estimated based on the analytical model in [55], using 45 nm technology metal specifications. In case of a single TSV, the estimated capacitance (7.7fF) is about 36% less than the capacitance of a TSV located within an island of four TSVs (about 12fF). However, if we consider four such single TSVs placed together, the estimated total TSV area (338.56 um^2) is about 23% higher than area of the 2x2 TSV island (275.56 µm^2). Therefore, optimizing the positions and distribution of TSVs during floorplanning has become a necessary step, to balance the inherent trade-off between TSV occupied area and TSV capacitance.

Table 4: List of notations used in this chapter

Notation	Definition
L_{wire}	total estimated wire length of a 3D net
r	TSV radius
κ	Number of device layers
N_{TSV}	Number of TSVs in a 3D net
p	Pitch between adjacent TSVs in a cluster
R_{TSV}	TSV dc resistance
TSV_{ref}	TSV subjected to maximum coupling capacitance in a cluster.
C_{TSV_max}	Capacitance of reference TSV in a cluster
C_{TW}	TSV-to-wire coupling capacitance
C_{TT}	TSV-to-TSV coupling capacitance
$Cc1$	Coupling with TSVs located orthogonally w.r.t TSV_{ref}
$Cc2$	Coupling due to TSVs located diagonally w.r.t TSV_{ref}
$\|\Phi\|$	TSV cluster capacity
Ψ	Set of TSV clusters that are fully assigned to nets.
$\emptyset_k$	Set of empty clusters on device layer ‘k’.
Cce	Capacitive coupling due to edge-effects.
max_Iter	Maximum number of floorplanning iterations given as input
$curr\_Iter$	Current Iteration

3.4. Delay Estimation in 3D Interconnects

A fast and reliable method of delay evaluation for 3D nets lies at the core of our delay-aware 3D floorplanning. According to [61], delay of a 3D net is impacted by the TSV dimensions (diameter and height), number of assigned TSVs and their positions along the length of the net. Our net-delay estimation based on Elmore-delay model, is well-suited for the early layout design stage, as the primary goal in this stage is to have a fast and fairly accurate delay estimation at a low computational cost. Fig.10 shows a two-pin 3D net spanning between two consecutive layers '*n*' and '*n+1*'. The net is assigned to a TSV located on the upper layer (*n+1*), at a distance '*x*' from the terminal pin 'p0'.

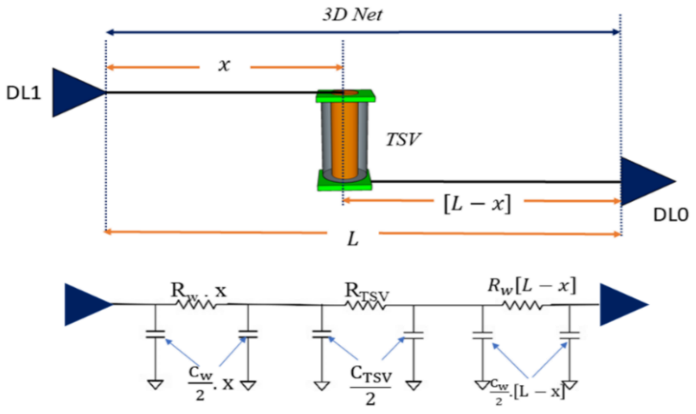


Figure 10. Elmore-model based RC delay estimation in 3D interconnects. A two-pin 3D net containing one TSV with corresponding distributed equivalent RC circuit is shown.

Considering 'L_{wire}' as the estimated length of the 3D net, the corresponding distributed RC equivalent model is shown. TSVs are replaced by their equivalent wirelength [47] and added to the actual length of the wire to calculate a net wirelength. The analytical expression for delay evaluation of a 3D net can be derived as described in [47] and shown in (1).

$$Net\ Delay = 0.5\ (R_w * C_w * L_{wire}^2 + R_{TSV} * C_{TSV} * {N_{TSV}}^2) \qquad (1)$$

Parameters R_{TSV} and C_{TSV} represent the TSV resistance (mΩ) and capacitance (fF) respectively. R_w and C_w denote the unit length resistance (mΩ/µm) and capacitance (fF/µm) of the wire. N_{TSV} represents the number of assigned TSVs to the net. It can be observed that net delay varies quadratically with length of the wire (L_{wire}) as well as the number of TSVs. Hence, for a short wire, L_{wire} being small, the net delay gets dominated by the TSV capacitance as compared to wire capacitance. This is the reason why interconnect delay is more effectively minimized by minimizing number of assigned TSVs considering the length of the wire, as compared to optimizing the total number of TSVs in the design [47].

3.4.1. Electrical Characterization of TSVs *(A) TSV Resistance* - The analytical expression for the dc resistance (R_{TSV}) of the TSV is given by (2):

$$R_{TSV} = \frac{\rho * H_{TSV}}{\pi r^2} \tag{2}$$

where, ρ is the resistivity of the TSV conducting material (taken as Cu = 1.68. x10^{-8} Ω.m). H_{TSV} is the height of the TSV and 'r' is the radius of TSV.

(B) TSV Capacitance - The TSV capacitance (C_{TSV}) is estimated using a fast and accurate analytical approach from [55]. TSV capacitance consists of two components as shown by (3):

$$C_{TSV} = C_{TW} + C_{TT} \tag{3}$$

C_{TW} represents the TSV-to-wire capacitance, which consists of components due to - i) coupling between top of TSV and bottom of overhead wires and ii) fringe capacitances between overhead wires and the sidewall of TSV [55]. Fig.11 depicts the simplified TSV-to-wire capacitance model as used in this work. The component C_{TT} represents the TSV-to-TSV capacitive coupling between adjacent TSVs in a cluster. C_{TT} is computed for a reference TSV, using analytical model as shown in (4).

$$C_{TT} = m * C_{C1} + n * C_{C2} \tag{4}$$

The component C_{c1} represents coupling due to 'm' number of TSVs located in four orthogonal direction w.r.t the reference TSV. Similarly, C_{c2} represents capacitive coupling from 'n' TSVs in diagonal direction w.r.t reference TSV.

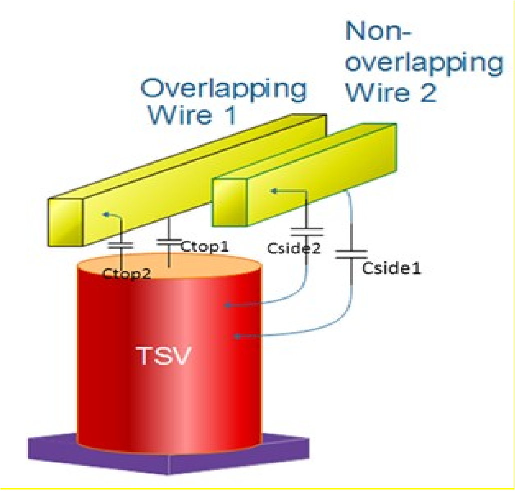


Figure 11. Components of TSV-to-wire capacitance (C_{TW}). Surface and fringe capacitances between top of TSV and sidewall of TSV with overhead wires are considered as per [55].

3.4.2 Assumption of worst-case TSV capacitance

Our goal during early layout design is to get a reasonably accurate evaluation without performing computationally intensive extraction of TSV parasitic capacitance. More accurate computation of TSV coupling capacitance is possible, and will be necessary, in the later stages, when the precise placement information of TSVs becomes available. Therefore, to ease the computational burden during floorplanning we assume the worst-case scenario, i.e., all TSVs within a cluster have the same capacitance which equals to the capacitance of the TSV that incurs the maximum coupling from its neighboring TSVs.

Fig.12(a) shows a 3x4 TSV cluster. The two middle-TSVs (highlighted yellow) represent the reference TSVs, as each of them is subjected to maximum coupling from eight adjacent TSVs - four TSVs in orthogonal direction and four TSVs in the diagonal direction. It can be noted that depending on the size of the cluster, the reference TSV can be a middle TSV, an edge TSV or a corner TSV as shown in Fig. 12(b) &(c) respectively.

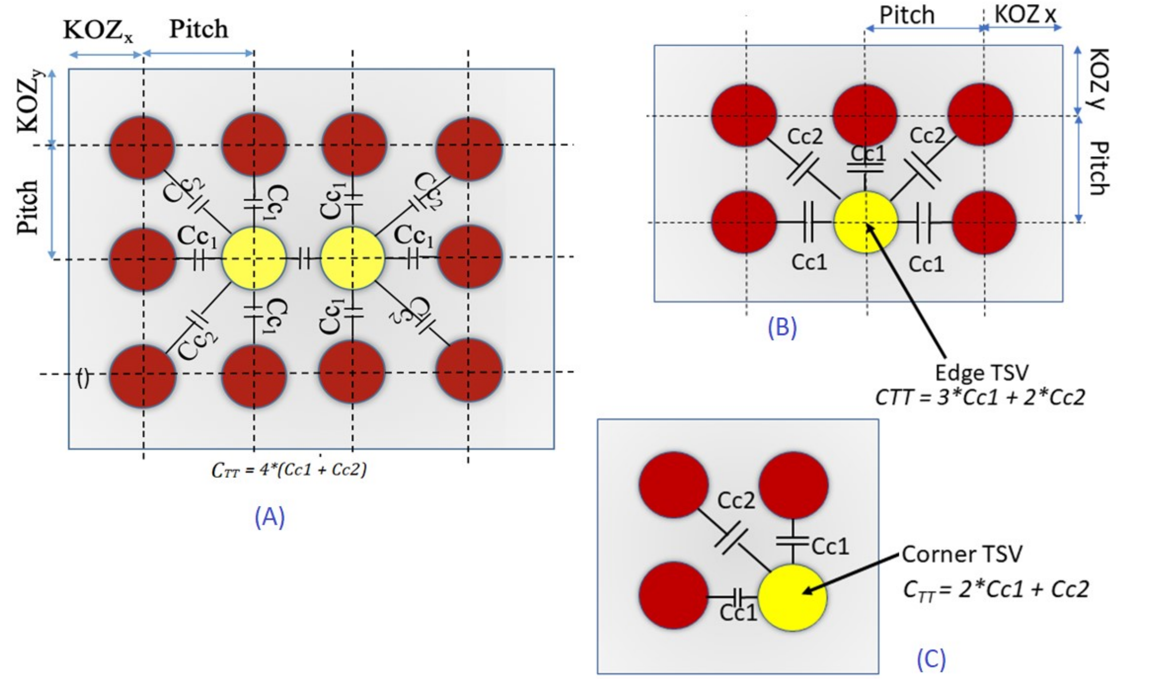


Figure 12. Reference TSVs (highlighted yellow) subjected to maximum capacitive coupling from adjacent TSVs within a cluster, are shown for three different TSV cluster types : (a) when reference TSV is a middle-TSV (b) when reference TSV is an edge-TSV (b) when reference TSV is a corner-TSV.

3.4.3. Consideration of TSV Edge-Effects

We have extended the analytical model for computing C_{TT} in (4), to include the non-negligible impact of edge-effects, due to presence of TSVs in surrounding clusters. This reflects the more realistic scenario where multiple TSV clusters may be placed together and are abutting on the layout. According to [106], the capacitance of a middle-TSV in a TSV cluster is not impacted by edge-effects due to the phenomenon of *Faraday cage-effect*. However, for those cluster sizes where the reference TSV is located on cluster's boundary, the impact of edge-effects due to adjacent clusters has to be included in computation of C_{TT}. For such clusters, the worst-case capacitance value will be higher than when they are placed as stand-alone clusters. Fig. 13(a)(b) & (c) show the coupling

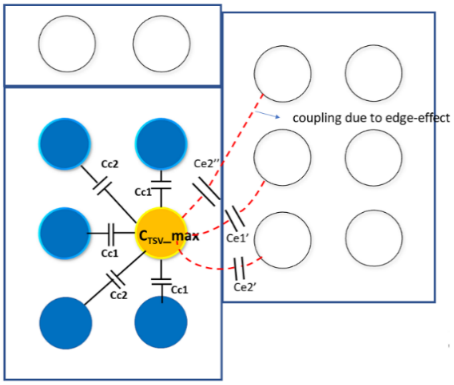


Figure 13(a) Illustration of TSV edge effects. Reference TSV (highlighted yellow) subjected to capacitive coupling from adjacent TSVs due to edge-effects. Coupling is influenced by relative positioning of the abutting cluster.

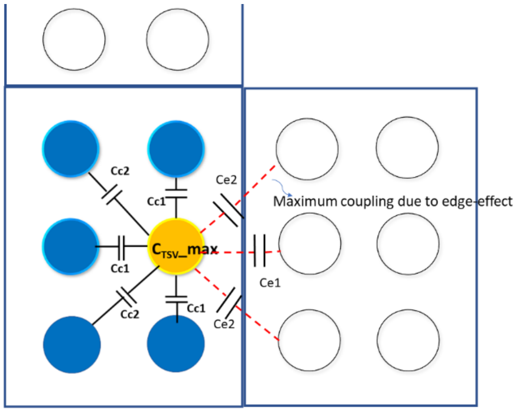


Figure 13(b). Maximum coupling due to TSV edge effects occurs when adjacent cluster is aligned as shown

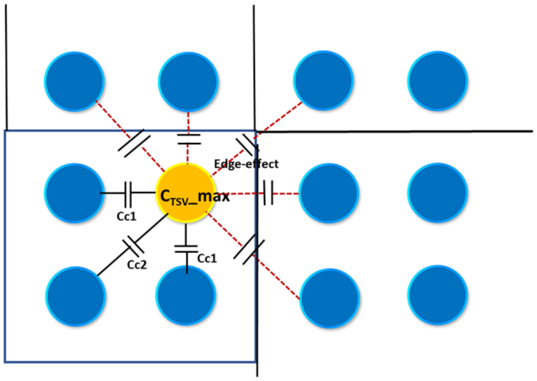


Figure 13(c). TSV edge-effects considering a corner TSV as reference TSV (note: only case of maximum coupling from nearby cluster is shown)

components due to edge effects from adjacent clusters in three scenarios. Fig.13(a)&(b) show the case when reference TSV is an edge TSV and Fig. 13(c) shows the case when reference TSV is a corner TSV. Fig. 13(b) shows our assumption of the worst possible scenario, where adjacent cluster is aligned in such a way that the reference TSV incurs the maximum coupling due to edge-effects from TSVs in the neighboring cluster.

In the scope of this work, the impact of non-neighboring TSVs on TSV-TSV coupling is considered negligible. For computation of C_{TT}, the impact of signal noise and cross-talk between adjacent TSVs is considered negligible and not included. It is assumed that C_{TT} includes the capacitance of the silicon substrate and the depletion region. Also, the influence of substrate resistance is ignored assuming a pure silicon substrate of very high resistivity. As we consider an operating clock frequency much lower than 2 GHz, the influence of TSV self-inductance is ignored in the scope of the paper [78].

3.5. Methodology of Dynamic TSV Clustering

We define the following terms as used in our approach.

i) TSV cluster – A TSV cluster ($\Phi_{m\,x\,n}$) is a regular array of one or more TSVs, defined by its capacity (number of TSVs), size (row x column) and aspect ratio (#of rows / #of columns). Fig.12(A) shows an example of a typical TSV cluster, with a capacity of twelve, size of (3x4) and aspect ratio of 0.75. The analytical model for area estimation of a TSV cluster is shown in (5):

$$Area(\varphi_{mxn}) = \left(2 * KOZ_x + p(n-1)\right) * \left(2 * KOZ_y + p(m-1)\right) \tag{5}$$

Parameters 'm' and 'n' represent the number of rows and columns in the cluster respectively. TSVs adjacent to each other in a row or column are separated by appropriate

TSV pitch '*p*'. The *TSV pitch* is the minimum required distance between TSVs in orthogon taken as twice of TSV diameter as recommended by the ITRS [4] to overcome TSV misalignment issues. Parameters KOZ_x and KOZ_y represent the required KOZ along the horizontal and vertical axes respectively. For an isolated single TSV, the size of KOZ is same on both x-axis and y-axis. With increase in number of TSVs in either vertical or horizontal direction, the required KOZ also increases, due to the cumulative stress interaction of TSVs [80]. The grouping of TSVs in a cluster, decreases the average TSV footprint, due to KOZ sharing amongst multiple TSVs [80]. TSV clusters are classified into the following types:

- *Single-TSV Cluster* – A cluster containing one TSV.
- *Empty Cluster* – A cluster that does not contain any TSV in it. An empty cluster does not contribute to the packing area, however it's topological relationship with other modules exists in the floorplan representation
- *Valid Cluster* – A cluster containing at least one unassigned TSV.

ii) Packing Area - The floorplan packing area is computed by multiplying the number of layers with the product of maximum estimated chip width and chip height amongst all device layers.

iii) *Effective Area Ratio* – The effective area ratio or EAR represents the percentage of the core area, actually utilized by all circuit modules. As shown in (6), EAR is defined as the ratio of the optimum area (total area consumed by all circuit blocks and TSV clusters) to the packing area. As the floorplan converges to an optimal solution, the packing area decreases and EAR increases.

$$EAR = \frac{optimum\ area\ (circuit\ blocks + TSV\ clusters)}{packing\ area\ \ (max\ \_chip\_w * max\_chip\_h*k)} \qquad (6)$$

3.5.1. Moves for Dynamic TSV Clustering

This section describes three new mutation operators or moves for floorplan perturbation. The moves for dynamic TSV clustering are selected probabilistically, resulting in stochastic grouping of TSVs on the layout. TSV clustering occurs intra-layer only i.e. performed on TSV clusters located on the same device layer. This implies that the number of TSVs on a device layer does not change after executing a move. It is worth emphasizing here that TSV clustering only impacts the capacity and size of the participating clusters, without affecting their individual topological positions on the floorplan. All moves are restricted by certain *conditions for feasibility*. An infeasible move is thus, identified at the floorplan representation level and eliminated without incurring any additional cost of computation.

i) Merge – Fig. 14 shows example of a merge operation. Two TSV clusters A and are chosen randomly on a device layer. The sum of their capacities is computed and gets assigned to one of the clusters selected randomly. This cluster is labeled as the merged cluster (cluster A shown in Fig. 14). For this merged cluster, a set of possible sizes (rows x columns) associated with its current capacity is obtained from a look-up table provided as input to the floorplanner. The optimal cluster size is chosen from the table, guided by

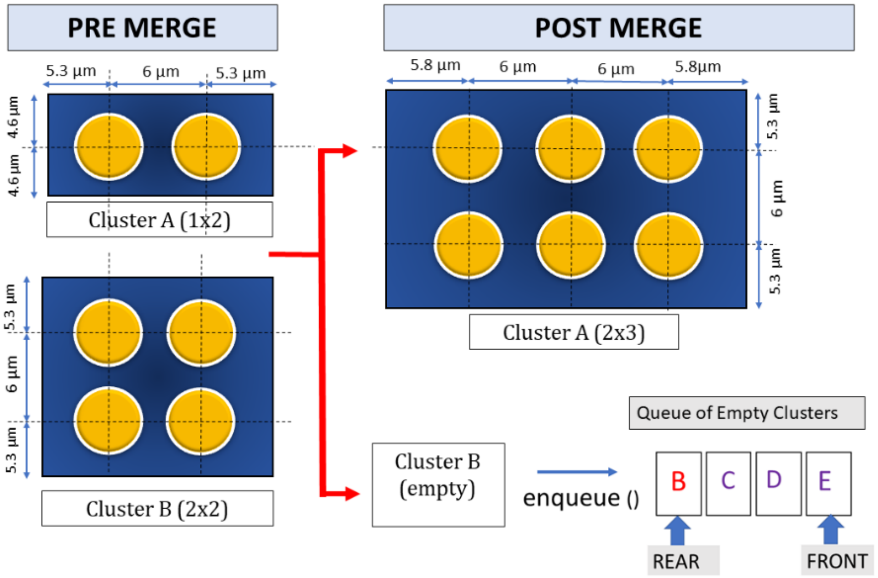


Figure 14. Merge of two clusters A and B. The area and capacity of merged cluster A is updated. Index of Cluster B joins the existing queue of empty clusters on the same device layer. Note: Labels - 'C', 'D', 'E', indicate the indices of empty clusters already present in the queue due to previous merge operations on the same device layer. A merge operation impacts the TSV footprint, packing area and decreases the population of clusters.

area evaluation of the current floorplan. Depending on the selected cluster size, the appropriate KOZ dimensions are computed using analytical model in [80]. The second cluster B that becomes empty, joins the queue of empty clusters on the same device layer.

Condition for Infeasibility of Merge – Let *φ1* and *φ2* represent the TSV clusters chosen on device layer '*k*' for merge. Then, a merge operation will be infeasible if:

$$(\varphi 1 \in \Theta_k) \,||\, (\varphi 2 \in \Theta_k)$$

where 'Θ_k' represents the set of empty clusters on device layer '*k*'.

ii) Division - Fig.15 shows the division of a TSV cluster A_{3x3}, chosen stochastically on a device layer. A set of possible combinations of cluster capacities, whose sum equals to the original cluster's capacity is constructed from a look-up table provided as input. A single combination is chosen stochastically. The original cluster A's capacity is updated to one of the capacity values in the selected combination. The second capacity value is allocated an empty cluster B, drawn from the queue as shown. Given the newly assigned

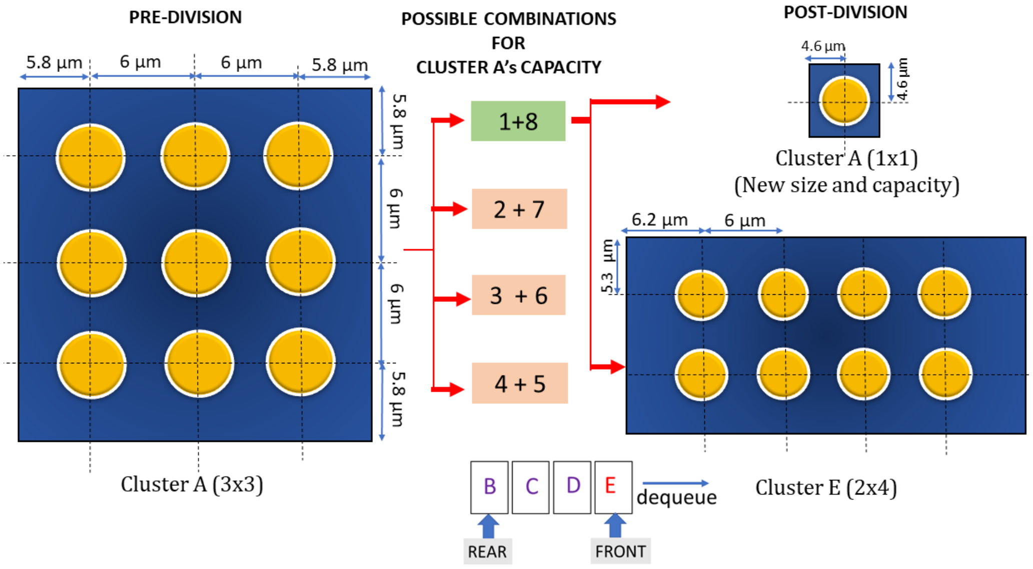


Figure 15. TSV cluster division. The original cluster A with nine TSVs is updated with a new capacity and size (1x1). The remaining TSVs are allocated to an empty cluster E (dequeued on the same layer). A division operation impacts the TSV footprint, total packing area and increases the population of clusters.

capacities, the appropriate size of each cluster is selected stochastically from a look-up table. The selection of sizes for resulting clusters is guided by the area evaluation of the current floorplan. Next, depending on their sizes, the required KOZ dimensions are computed according to [80] and applied to both the clusters.

Conditions for Infeasibility of Division - Let φ represent the TSV cluster selected for division. Let $|\varphi 1|$ and $|\varphi 2|$ represent the capacities of the clusters resulting after division. The division operation will be infeasible if any of the conditions hold true:

1. $\varphi \in \emptyset_k$, where $\emptyset_\mathrm{k}$ represents the queue of empty clusters on 'k$^{\mathrm{th}}$' device layer.
2. *(|φ1| =1)* i.e., a single-TSV is selected for division.
3. {*(*|*φ1*| = 0) || (|φ2| = 0)} i.e., a cluster is empty after division
4. $\emptyset_\kappa$ = *Φ*. i.e., the selected device layer 'κ' does not contain an empty cluster required for TSVs' allocation during after division.

i) Recombination – Fig.16 shows example of a cluster recombination. Two clusters A_{2x2} and B_{2x2} are chosen stochastically on a device layer. The sum of their capacities is computed. As in the case of division operation, a set of possible combinations of cluster capacities, whose sum equals to the total capacity is drawn from a reference table. One of combinations is selected randomly and the new capacity values are applied to the original clusters. Based on the newly assigned capacities, the new sizes for the original clusters are chosen from look-up table guided by floorplan's area evaluation. As shown, cluster A updates from its original size (2x2) to (1x2) while cluster B changes to size of (2x3). It is notable that, when the capacities of the participating clusters differ from their respective original capacities, then the TSV footprint is impacted due to updated KOZ dimensions. In case the participating clusters simply swap their respective capacities, then, the total TSV area remains unchanged. However, the packing of modules will get impacted in the subsequent iteration.

Conditions for Infeasibility of Recombination - Let |*φ1*| and |*φ2*| represent the original capacities of the two selected clusters *φ1* and *φ2* respectively. |*rφ1*| and |*rφ2*| represent the updated capacities of *φ1* and *φ2* after recombination. A recombination is *infeasible* if any of the conditions hold true:

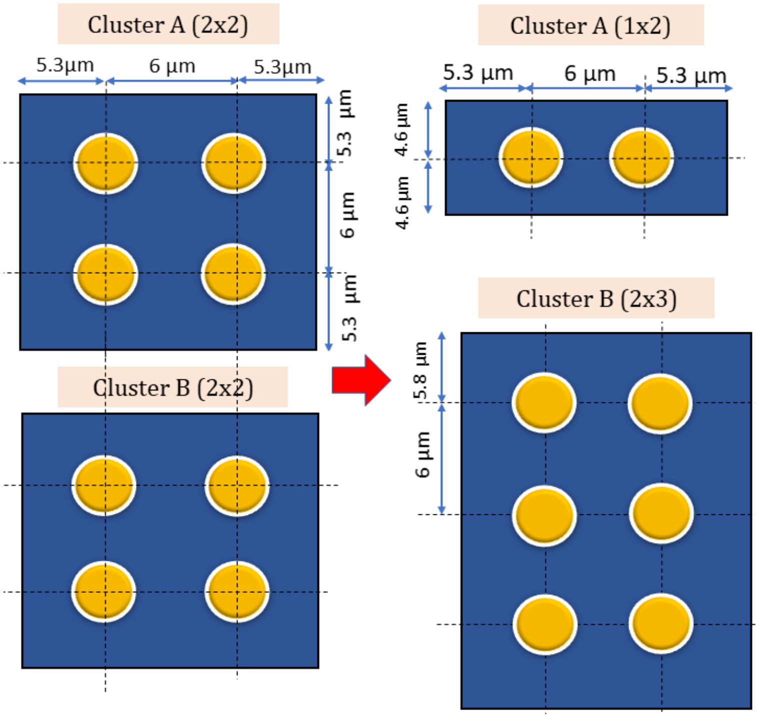


Figure 16. Illustration of recombination. Two clusters A and B recombine to form clusters with new capacities and sizes. Both TSV footprint and packing area are impacted due to recombination while total population of clusters remains the same.

1. $(|\varphi 1| = 1)$ && $(|\varphi 2| = 1)$, i.e., chosen clusters are single.

2. $(r\varphi 1 \in \emptyset_{\kappa})$ || $(r\varphi 2 \in \emptyset_{\kappa})$ i.e. either of the two clusters becomes empty after recombination.

3. $(|\varphi 1| = |r\varphi 1|)$ && $(|\varphi 2| = |r\varphi 2|)$ i.e., the updated capacities of clusters are same as their respective original capacities.

3.5.2. Probabilistic Selection of Clustering Moves

TSV clustering moves are selected based on their assigned probability of occurrence as shown in Table 5. In our approach, the probability of each move varies dynamically depending on - (i) packing area of current floorplan (Refer sec. 3.5) and (ii) the stage of floorplanning. Our floorplanning is divided into three stages as below:

Table 5. Probability of Occurrence assigned to each clustering move. The assigned value depends on stage of floorplanning and evaluated effective area ratio of floorplan.

Conditions for Move Selection	*Set of Probability Values*	*TSV Clustering Move*	*Key*	*Probability (ρ)*
EAR ≤ 0.72 OR Stage I	list 1	Recombination	0	0.005
		Division	1	0.005
		Merge	2	0.99
0.72< EAR ≤ 0.8 OR Stage II	list 2	Recombination	0	0.01
		Division	1	0.05
		Merge	2	0.94
EAR > 0.80 OR Stage III	list 3	Recombination	0	0.30
		Division	1	0.35
		Merge	2	0.35

1. Stage I – 0th iteration to 3/5th of maximum iterations

2. Stage II- From 3/5th of maximum iterations to 4/5th of maximum iterations

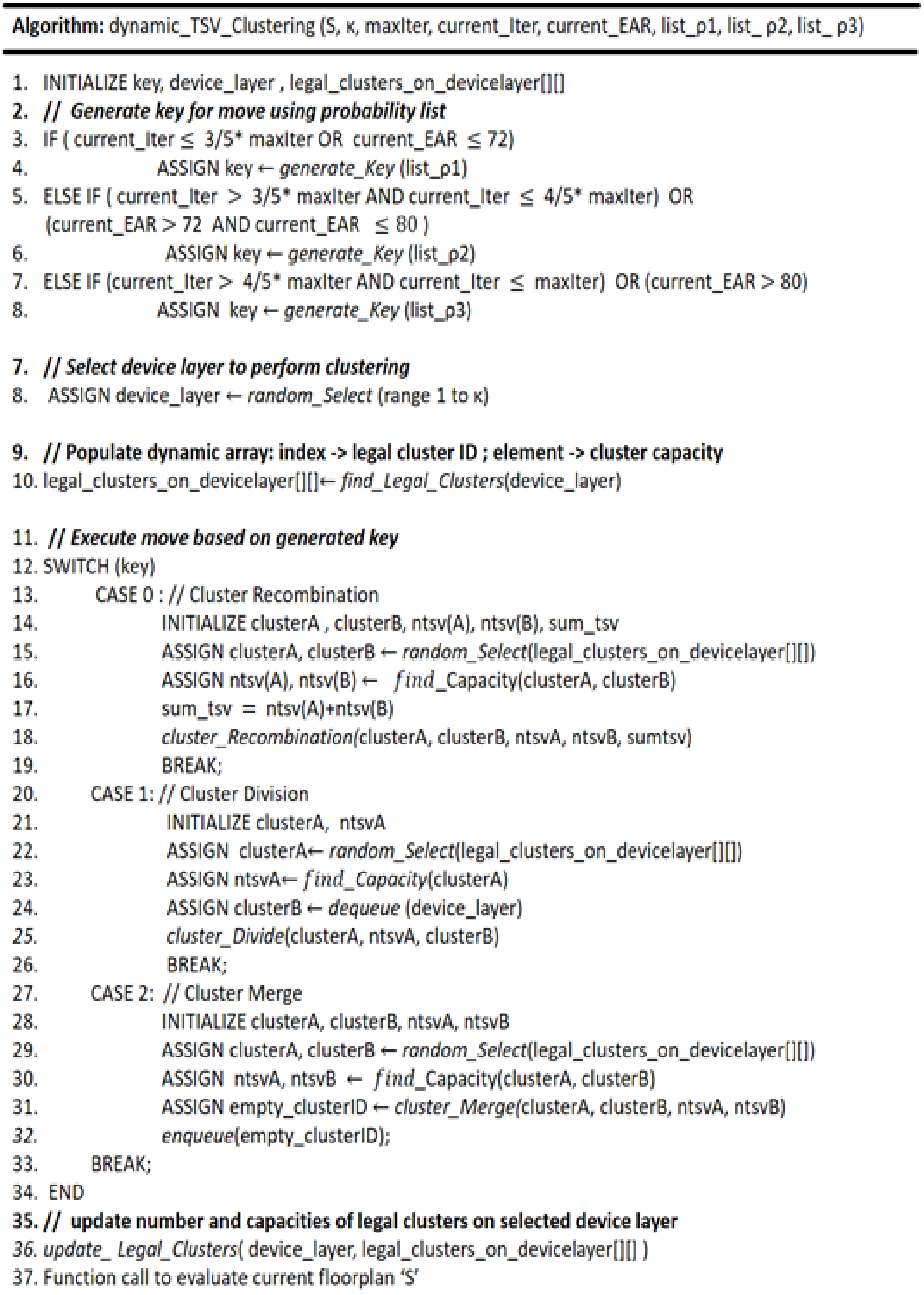
3. Stage III- From 4/5th of maximum iterations to end of iterations or termination of floorplan run, whichever should occur first.

The values of probability assigned to each move are derived after running various tests with different values during the initial development of the algorithm, as it is done when developing probabilistic algorithms. As there is no way to establish the best sets of weights/probabilities in non- deterministic algorithms, we selected the applied probabilities based on packing efficiency of the final solution during the initial runs. The maximum number of iterations is set based on our initial tests to see when a solution converges, i.e., the cost attains an almost constant value for a certain user-specified number of iterations.

In scope of this work, we have assumed a bottom-up clustering approach, i.e., all clusters are single-TSV clusters at the beginning of floorplanning. Therefore, to increase the number of feasible moves in stage I, the probability of merge is set at highest value (99%), while the probability of division and recombination is kept significantly low. The presence of a large number of single TSVs makes the division infeasible. A recombination between two single-TSVs is also infeasible, as it will generate two single-TSVs, without having any impact on the TSV area. With repeated merge operations in stage I, a number of large-sized TSV clusters become available in Stage II. The probability of occurrence for division is raised at this stage, which allows the formation of medium-sized and small-sized TSV clusters. Our initial experiments reveal that an increased probability of division at this stage provides critical improvement in packing efficiency due to redistribution of smaller clusters amongst large and medium clusters. In stage III, the probability of recombination is significantly raised to further optimize the sizing of TSV clusters to improve area.

3.5.3. Steps of Dynamic TSV Clustering Algorithm

The steps of the proposed dynamic TSV clustering algorithm are shown in Fig.17. The inputs are– current floorplan ‘S’, ‘κ’ device layers, maximum number of iterations (*max_Iter*), current iteration *(current_Iter),* EAR of the current floorplan (*current_EAR*) and the probabilities for clustering moves for different floorplanning stages. The problem of dynamic TSV clustering is to select the appropriate clustering move which simultaneously satisfies the conditions of probability of occurrence and conditions of feasibility of the move, while minimizing the floorplan packing area.



```
Algorithm: dynamic_TSV_Clustering (S, κ, maxIter, current_Iter, current_EAR, list_ρ1, list_ ρ2, list_ ρ3)

INITIALIZE key, device_layer , legal_clusters_on_devicelayer[][]
// Generate key for move using probability list
IF ( current_Iter ≤ 3/5* maxIter OR current_EAR ≤ 72)
                ASSIGN key ← generate_Key (list_ρ1)
ELSE IF ( current_Iter > 3/5* maxIter AND current_Iter ≤ 4/5* maxIter) OR
(current_EAR > 72 AND current_EAR ≤ 80 )
                 ASSIGN key ← generate_Key (list_ρ2)
ELSE IF (current_Iter > 4/5* maxIter AND current_Iter ≤ maxIter) OR (current_EAR > 80)
                ASSIGN key ← generate_Key (list_ρ3)

// Select device layer to perform clustering
 ASSIGN device_layer ← random_Select (range 1 to κ)

// Populate dynamic array: index -> legal cluster ID ; element -> cluster capacity
legal_clusters_on_devicelayer[][]← find_Legal_Clusters(device_layer)

// Execute move based on generated key
SWITCH (key)
        CASE 0 : // Cluster Recombination
                INITIALIZE clusterA , clusterB, ntsv(A), ntsv(B), sum_tsv
                ASSIGN clusterA, clusterB ← random_Select(legal_clusters_on_devicelayer[][])
                ASSIGN ntsv(A), ntsv(B) ←  find_Capacity(clusterA, clusterB)
                sum_tsv = ntsv(A)+ntsv(B)
                cluster_Recombination(clusterA, clusterB, ntsvA, ntsvB, sumtsv)
                BREAK;
        CASE 1: // Cluster Division
                 INITIALIZE clusterA, ntsvA
                 ASSIGN clusterA← random_Select(legal_clusters_on_devicelayer[][])
                 ASSIGN ntsvA← find_Capacity(clusterA)
                 ASSIGN clusterB ← dequeue (device_layer)
                 cluster_Divide(clusterA, ntsvA, clusterB)
                 BREAK;
        CASE 2: // Cluster Merge
                INITIALIZE clusterA, clusterB, ntsvA, ntsvB
                ASSIGN clusterA, clusterB ← random_Select(legal_clusters_on_devicelayer[][])
                ASSIGN ntsvA, ntsvB ← find_Capacity(clusterA, clusterB)
                ASSIGN empty_clusterID ← cluster_Merge(clusterA, clusterB, ntsvA, ntsvB)
                enqueue(empty_clusterID);
        BREAK;
 END
// update number and capacities of legal clusters on selected device layer
update_ Legal_Clusters( device_layer, legal_clusters_on_devicelayer[][] )
Function call to evaluate current floorplan 'S'
```

Figure 17. Steps of dynamic TSV clustering algorithm

i. *Stage I: Select Move* – In the first phase, we generate an appropriate key associated with a move. Based on the current_Iter and current_EAR, the correct list of probability is selected from Table 5. A utility function generate_Key uses this list to compare against a random number 'r'. For a key to be selected, 'r' should be less than the key's probability value.

ii. *Stage 2: Select Device Layer* – In this step, a valid device layer is selected stochastically, between the range (1 to κ-1). The bottom-most layer is excluded, as it does not contain TSVs. A device layer is valid if the chosen move in stage I is feasible on it (please refer Sec IV-B). Once a valid tier is found, a utility function find_TSV_clusters() is used to store the information of all non-empty TSV clusters on the layer with their associated capacities. Empty TSV clusters are stored in a separate queue.

iii. *Stage 3: Execute Move* – Depending on the selected move, the candidate TSV clusters are given as input to the corresponding function to execute the move as described in Sec. 3.5.2

3.6. Design Flow of 3D Floorplanning with Dynamic TSV Clustering (3D-DTC)

The design flow of our 3D floorplanning with dynamic TSV clustering (3D-DTC) is shown in Fig. 18. Our floorplanner is based on evolutionary algorithm (EA) and uses sequence pair (SP) representation [34]. The inputs to floorplanner consist of: a) list of circuit blocks b) netlist b) number of device layers c) TSV specifications and e) metal specifications depending on the technology node. The floorplanner performs simultaneous placement of circuit blocks with TSVs, while optimizing the desired performance metrics – area, total interconnect delay. For a given circuit, the maximum needed TSVs is estimated based on a probabilistic model as shown in [47].

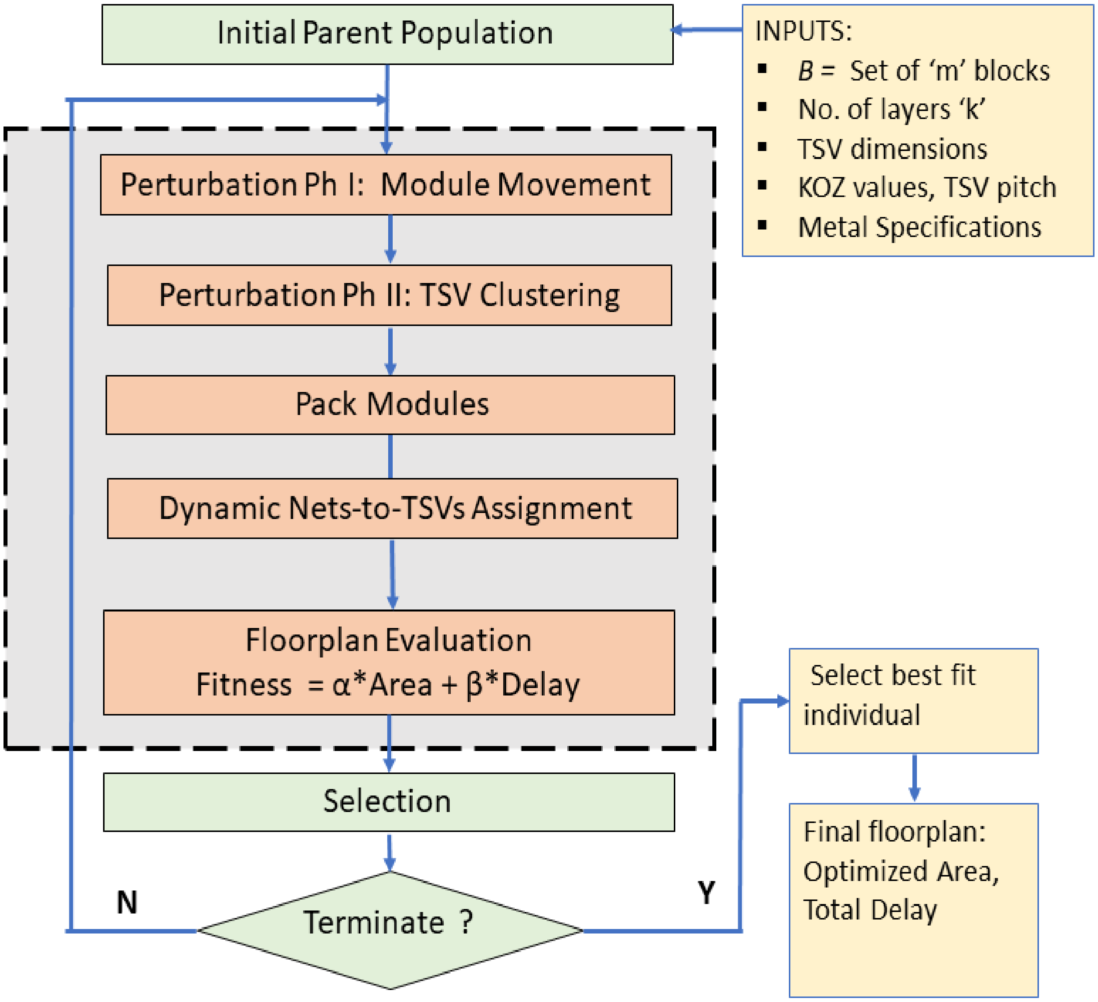

Initial Parent Population
INPUTS:
▪ *B* = Set of 'm' blocks
▪ No. of layers 'k'
▪ TSV dimensions
▪ KOZ values, TSV pitch
▪ Metal Specifications
Perturbation Ph I: Module Movement
Perturbation Ph II: TSV Clustering
Pack Modules
Dynamic Nets-to-TSVs Assignment
Floorplan Evaluation
Fitness = α*Area + β*Delay
Selection
N
Terminate ?
Y
Select best fit individual
Final floorplan: Optimized Area, Total Delay

Figure 18. Design flow of 3D floorplanning with dynamic TSV clustering Area shaded in grey is repeated for each candidate floorplan in a population.

1. *Initial population* - An initial parent population is created. Each individual in parent population is a unique solution, generated by random distribution of circuit blocks and estimated TSVs amongst the device layers.

2. Solution Perturbation - Parents are subjected to perturbation or stochastic "reproduction" resulting in offsprings. A floorplan is perturbed by a move, selected probabilistically from a set of mutation operators. A floorplan is perturbed in two phases:

- **Inter/Intra-layer Module Movement**- An operator for module movement is selected from a set of seven mutation operators as defined in [35]. Both intra-layer and inter-layer movements of modules are allowed to occur. These moves are responsible for altering the relative topological positions on the layout. Therefore, they have an impact on the floorplan packing area without affecting the TSV occupied area.
- **TSV Clustering Move** – In the second phase, a TSV clustering move is executed as described in Sec. 3.5.3.

3. *Nets-to-TSVs Assignment* – A floorplan perturbation is followed by estimation of packing area and computation of EAR as shown in (6). Upon reaching a desired minimum EAR, the nets-to-TSVs assignment begins. Fig.19 shows the steps for our dynamic nets-to-TSVs assignment algorithm. Given a current floorplan '*S*', netlist '*N*', TSVs' parameters and technology specifications, the problem is to assign each 3D net spanning between two contiguous tiers, a TSV on the upper tier such that delay contributed by a TSV to the net is minimized [47]. Prior to assignment, the set of all valid TSV clusters on each device layer '*k*' are stored in a map 'V_k'. A TSV cluster is valid if it contains at least one TSV available for assignment. Each element in the map has its unique index and the associated cluster capacity. Before assignment, nets are stochastically selected from the netlist '*N*'. This is done to enable a solution that is independent of the order of selection of nets. For a two-pin 3D net *'i'*, the required number of TSVs (ntsvi) is computed based on its terminal pin locations (l_{max_i} and l_{min_i}) on device layers. A TSV cluster is selected stochastically from

‘V_k’ on each device layer. The selection of a TSV cluster is guided by delay evaluation of the net as shown in (1). From each chosen cluster, a TSV is assigned to net ‘*i*’. The number of TSVs assigned in each valid TSV cluster is stored in a separate array ‘U_k’. After each assignment, the associated values in array ‘U_k’ is updated.

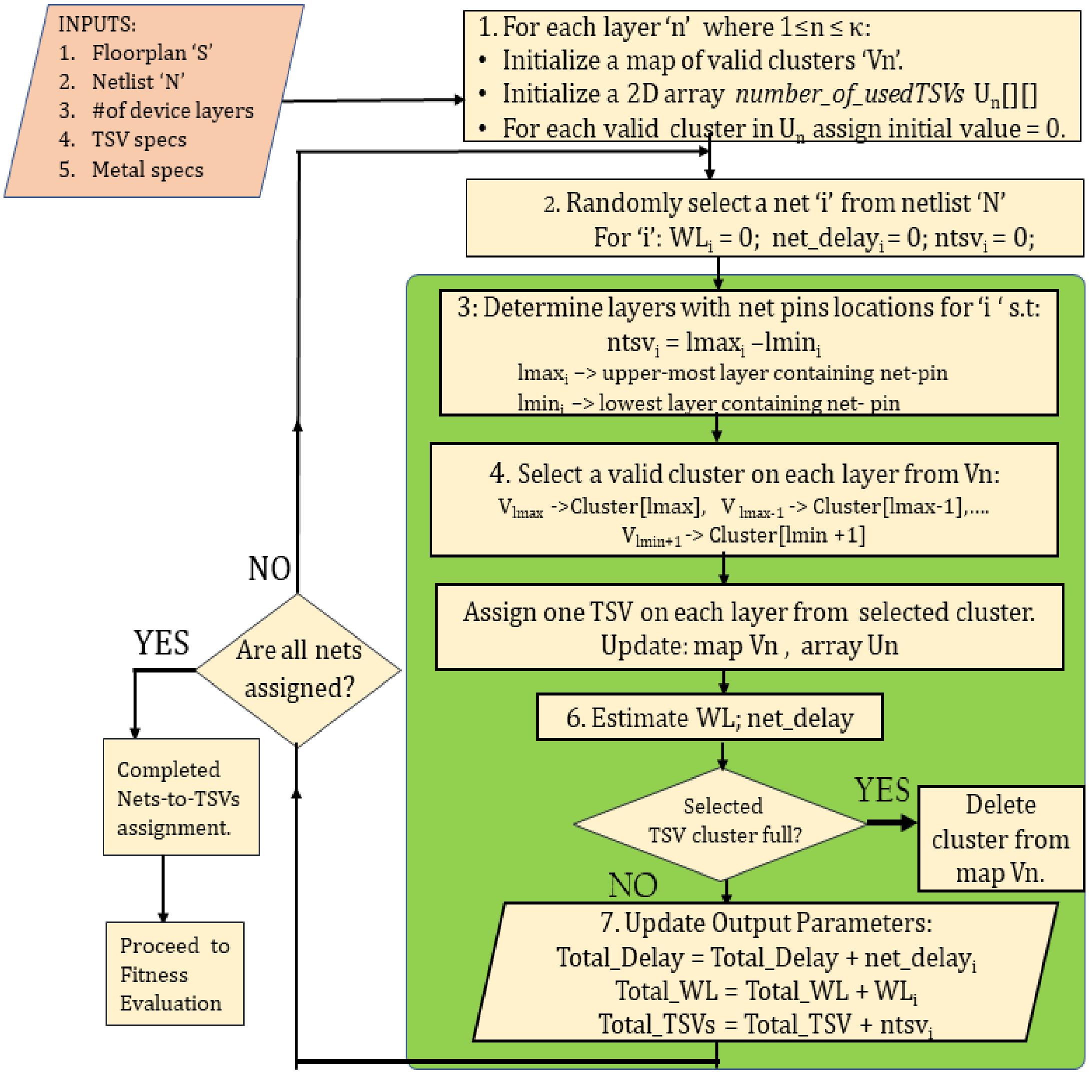

INPUTS:
1. Floorplan ‘S’
2. Netlist ‘N’
3. #of device layers
4. TSV specs
5. Metal specs
1. For each layer ‘n’ where 1≤n ≤ κ:
• Initialize a map of valid clusters ‘Vn’.
• Initialize a 2D array number_of_usedTSVs Un[][]
• For each valid cluster in Un assign initial value = 0.
2. Randomly select a net ‘i’ from netlist ‘N’
For ‘i’: WLi = 0; net_delayi = 0; ntsvi = 0;
3: Determine layers with net pins locations for ‘i ’ s.t:
ntsvi = lmaxi –lmini
lmaxi -> upper-most layer containing net-pin
lmini -> lowest layer containing net- pin
4. Select a valid cluster on each layer from Vn:
Vlmax ->Cluster[lmax], V lmax-1 -> Cluster[lmax-1],....
Vlmin+1 -> Cluster[lmin +1]
Assign one TSV on each layer from selected cluster.
Update: map Vn , array Un
6. Estimate WL; net_delay
Selected
TSV cluster full?
YES
Delete
cluster from
map Vn.
NO
7. Update Output Parameters:
Total_Delay = Total_Delay + net_delayi
Total_WL = Total_WL + WLi
Total_TSVs = Total_TSV + ntsvi
NO
YES
Are all nets
assigned?
Completed
Nets-to-TSVs
assignment.
Proceed to
Fitness
Evaluation

Figure 19. Steps of dynamic nets-to-TSVs assignment. The steps within shaded area are repeated for each 3D net in the input netlist

A cluster is removed from map '*Vk*' as soon as it becomes invalid. In this respect, usage of hash-map structure allows very fast search and delete operations in *O(1)* time, irrespective of the size or complexity of the circuit. Each assignment of a net is followed by estimation of wirelength. The wirelength is estimated using an accurate net-splitting method by considering the coordinates of TSVs along the wire on each device layer [44]. This is followed by estimation of net delay according to analytical model in (1). The process (highlighted in green) is repeated for all 3D nets, until assignment is complete. The output parameters - total delay, wirelength and number of used TSVs are provided for the next step i.e. fitness evaluation.

4. *Fitness Evaluation* – The fitness of a floorplan is evaluated using a pre-defined cost function. For our experiments, we have considered two different cost functions – CF1 and CF2 as shown in (7) and (8) respectively. α, β, γ and ρ are tuning parameters assigned to the cost metrics.

$$CF1 = \rho * area + \alpha * WL + \beta * N_{TSV} \qquad (7)$$

$$CF2 = \rho * area + \gamma * total\ delay \qquad (8)$$

CF1in (7) represents a wirelength-aware cost function that has two separate components - i.e. wirelength (WL) and the total number of TSVs (N_{TSV}). Therefore, CF1 does not optimize delay directly. CF2 is a delay-aware cost function which directly optimizes the total interconnect delay [34].

5. *Termination* – The criteria for termination of floorplan is based on when the solution converges, i.e., the fitness value repeats for a user-specified number of iterations.

Upon termination, the best fit individual in the final population is selected as the optimized floorplan.

Table 6. Technology and TSV specifications used for experiments in this chapter

Device Technology	45 nm
Die-bonding Type	Front-to-Back (F2B)
TSV Process Technology	Via -middle
TSV Diameter	3 µm
TSV Aspect Ratio (*H/W*)	10:1
TSV Material Resistivity (mΩ.um)	Cu = 17.2
TSV Pitch	2* TSV Diameter
Clock Frequency (*f*)	1 GHz
Metal Pitch	280 nm
Metal Width	140 nm
Metal Thickness	252 nm
Metal Spacing	140 nm
Wire Resistance per unit length (R_w)	0.439 Ω/µm
Wire Capacitance per unit length (C_w)	0.171fF/µm

3.7. Experimental Results

We implemented our 3D floorplanning with TSV clustering (3D-DTC) in C++/STL. The experiments were performed on a 4xDual Core Sun SPARC IV CPUs at 1.35 GHz andtotal 32 GB RAM. We consider the three largest Gigascale Systems Research Center (GSRC) benchmarks whose specifications are shown in Table 2. We assume a fixed-chip outline floorplan with four device layers. For each tier, the aspect ratio is set close to one and a maximum 10% whitespace is allowed. Table 6 shows the TSV parameters and technology specifications used for our experiments. A via-middle TSV technology is assumed, as it is more established than other TSV processes. Unlike the via-first approach, the use of common materials as TSV filler is allowed in via-middle process. Also, unlike via-last TSVs, the via-middle TSV does not interfere with the metal layer. All TSVs in this work are assumed as signal TSVs, as they are not strictly required to be vertically stacked. For wire dimensions, we consider Intel's 45 nm technology specifications for intermediate (semi-global level) metal layers. These metal levels are commonly used to route signals between consecutive device layers through TSVs.

3.7.1. Optimization of TSV Cluster Sizes and Floorplan Packing Area

Fig.20 shows for n100 benchmark, the variation in number of TSV clusters on each device layer as the floorplan proceeds to convergence. In our approach, a floorplan converges when the fitness value repeats for a certain number of times as specified by the user. The vertical axis shows number of clusters formed on top-most device layer (DL3) and two intermediate device layers - DL2 and DL1. The horizontal axis represents the stages of floorplanning from iteration #100 to the last iteration #66,000. Table 7 shows the variation in the floorplan packing area corresponding to each iteration. The results for

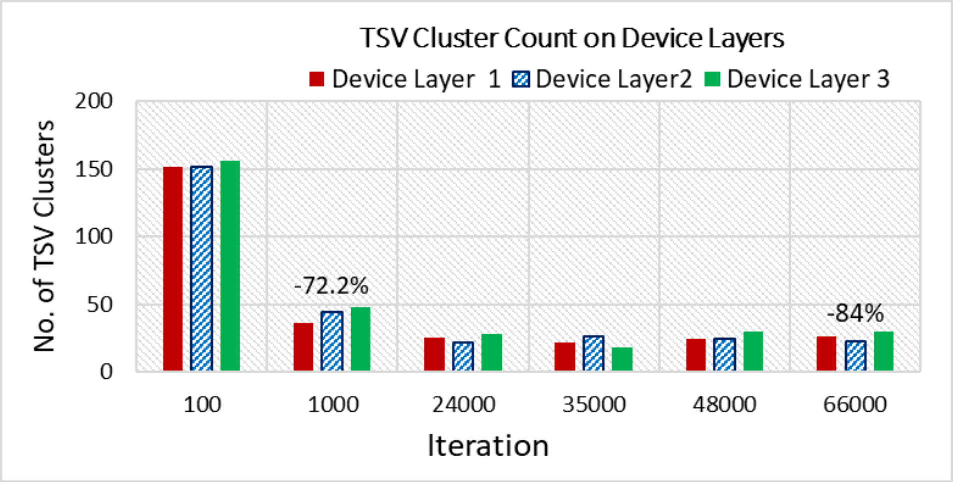


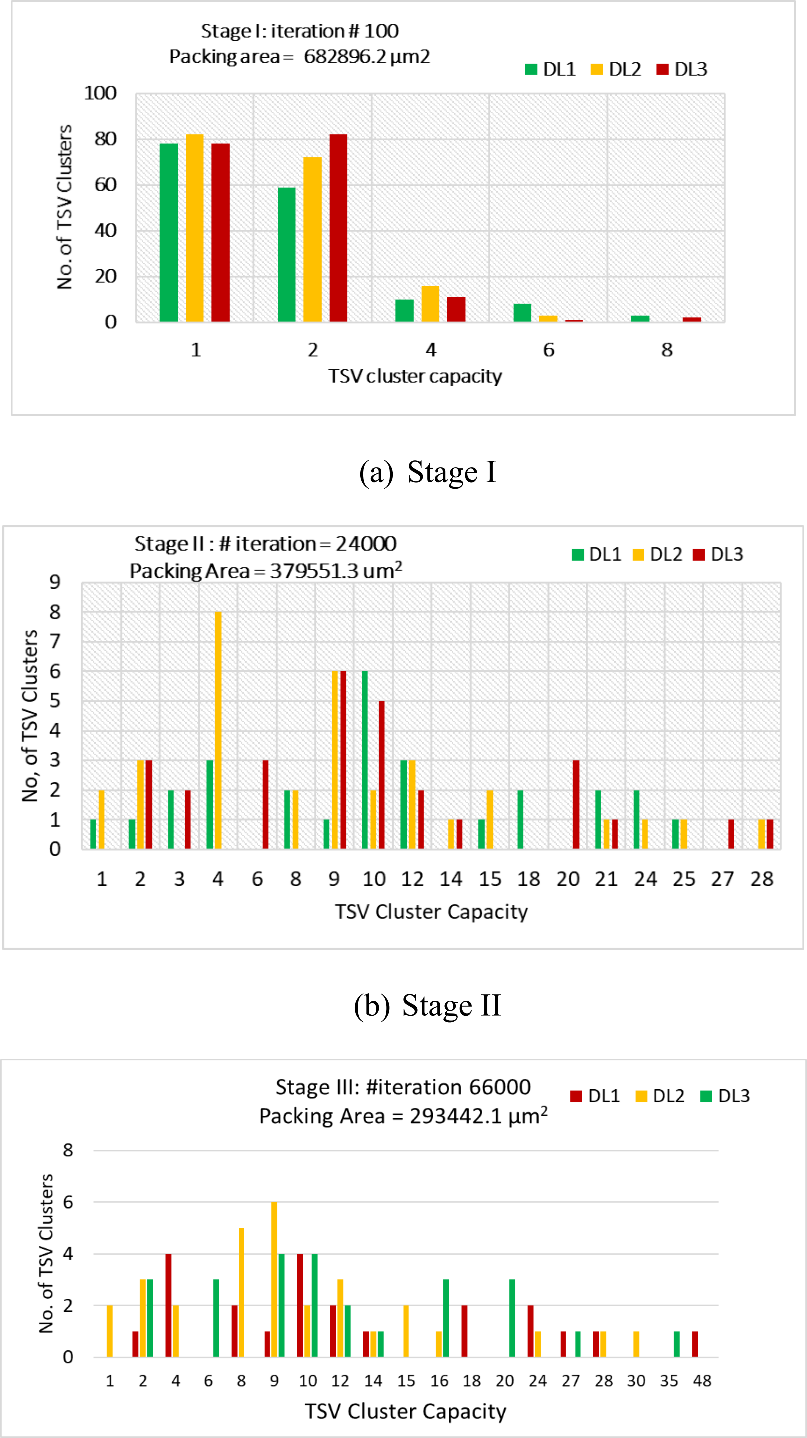
Figure 20. Variation of number of TSV clusters on top-three device layers with iterations. Data labels indicate percentage reduction in TSV cluster count as compared to base case – iteration #100. (For GSRC benchmark: n100).

Table 7. Variation in floorplan packing area through iterations (GSRC benchmark: n100)

# Iteration	Floorplan Packing Area (μm^2)
100	682896.2
1000	554137.6
24000	379551.3
35000	331649.3
48000	312741.4
66000	293442.1

n200 circuit are shown here, as the results for n100 and n300 show a similar trend. In the initial stages (iteration 100-1000) due to highest probability of merge, the total cluster count on all device layers drops dramatically by about 72%. As TSV area shrinks with large number of merged clusters, the packing area drops significantly by 26%. Interestingly, in stage II the packing area improves by about 8%, even as the total cluster count increases by 18% due to rise in probability of division move. This phenomenon is attributed to the rise in the number of small and medium-sized clusters, which helps in improving the overall packing efficiency by utilizing deadspace. In the final stages of floorplanning (> 48000th iteration), the probability of merge and divide become relatively smaller than prior stages.

Therefore, it is observed that the resultant impact on the total cluster count on the device layers is rather trivial at this stage. However, due to higher probability of recombination in this stage, the TSVs get redistributed amongst the existing clusters, further optimizing their capacities and sizes. Further improvement in area is observed due to efficient packing of small, medium and large-sized clusters on the layout. Our results with benchmarks n200 and n300 also show a similar trend.



(a) Stage I

(b) Stage II

(c) Stage III

Figure 21. Dynamic variation in distribution of cluster sizes on three device layers at three stages of floorplanning. With formation of a wide range of cluster sizes, packing area improves by 56% from (a) stage I to (c) stage III. GSRC benchmark(n100).

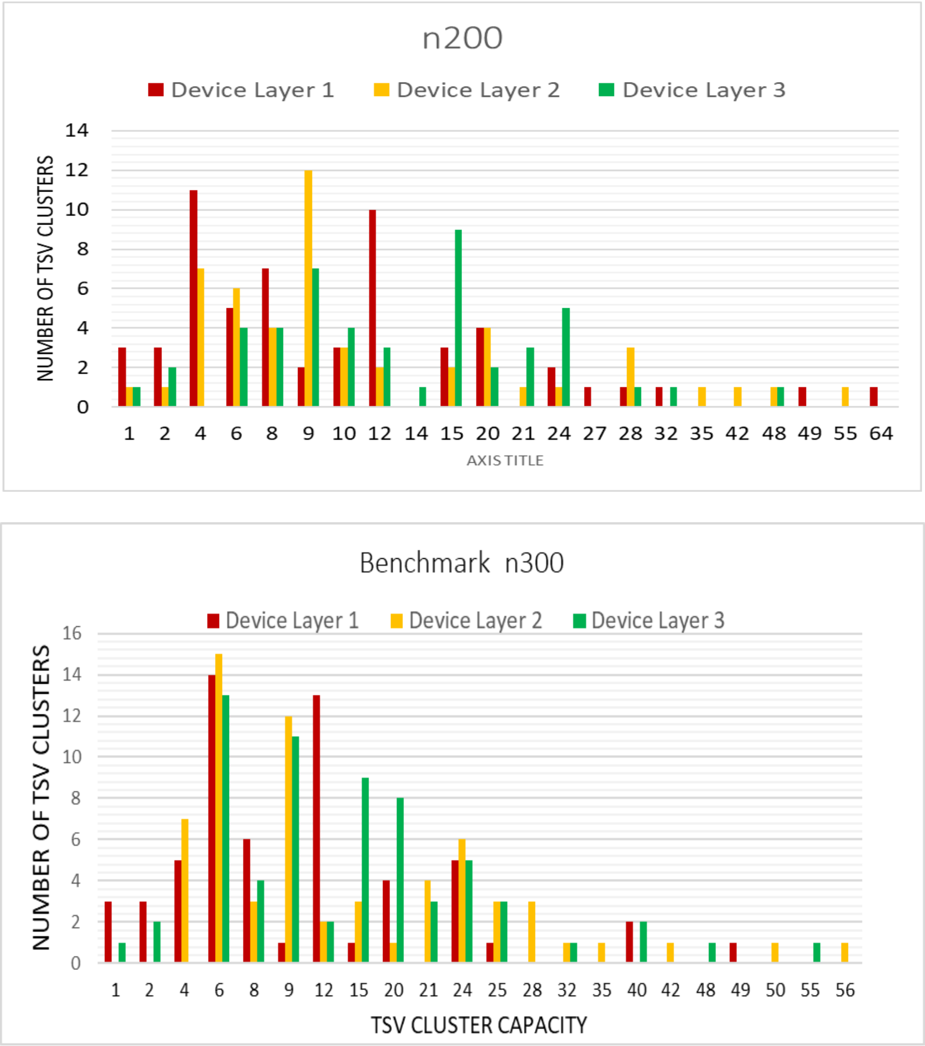


Figure 22. Distribution of cluster sizes on final optimized floorplan for GSRC benchmarks: n200 (top) and n300 (bottom)

Fig. 21(a)(b) and (c) show the variation in the distribution of cluster sizes amongst the three top device layers at all three stages of floorplanning. As the floorplan converges at stage III, TSVs are optimally distributed amongst a wide spectrum of cluster sizes. This results in improvement of the floorplan packing area by 56% at stage III.

Fig. 22 shows the trend of distribution of cluster sizes on final floorplan for benchmarks n100 and n300 respectively.

3.7.2. Comparison of Area Efficiency with Other Approaches

In this experiment, we compare the area efficiency (EAR) of the proposed floorplanning (3D-DTC) method with two commonly used 3D floorplanning approaches – i) co-placement of circuit modules with single TSV cells and ii) co-placement of circuit modules with a fixed number and size of TSV islands. In the second case, we distribute the estimated TSVs equally amongst twenty-four TSV islands (i.e., eight TSV islands are located on each device layer except for the bottom-most device layer).

Fig.23 compares the EAR of all three approaches. Compared to 74% EAR achieved with single TSVs approach and 86.5% EAR with fixed-sized TSV islands, the TSV clustering approach achieves an EAR of 91.7%. According to (6), two key factors – the total TSV area and the packing area, impact the overall EAR. With dynamic TSV clustering, both the abovementioned factors are simultaneously impacted with this approach, unlike other two approaches where TSV area remains unchanged. It is interesting to observe that with fixed-sized TSV islands, the EAR gets nearly saturated towards the final stages of floorplanning (>20000th iteration). However, with TSV clustering, the EAR continues to improve past 20,000th iteration. This is due to better packing efficiency is achieved as more large and medium-sized clusters undergo recombination at this stage.

3.7.3. Impact of TSV Clustering without KOZ sharing

We investigated the impact of dynamic TSV clustering on area efficiency without considering the contribution of shared KOZ during cluster formation. For this experiment we consider three cases for comparison – i) floorplanning with a fixed number and size of TSV islands (base-case) ii) dynamic TSV clustering with KOZ sharing within clusters and

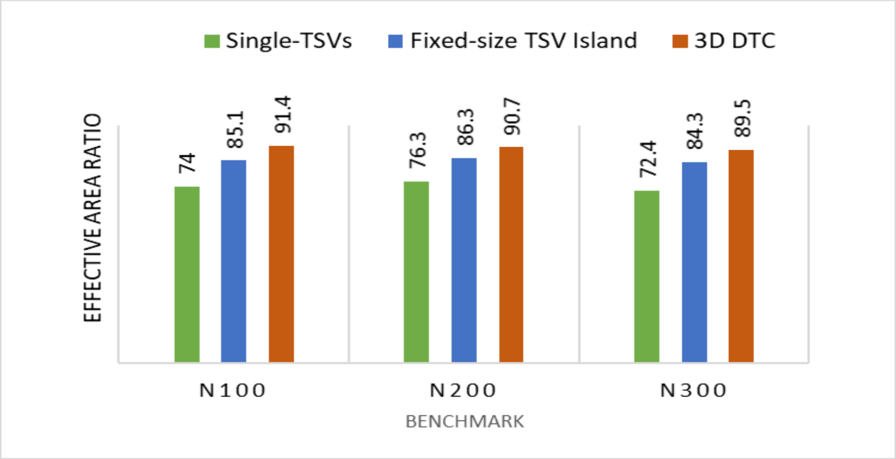


Figure 23. Effective area ratio (EAR) achieved with proposed approach (3D-DTC) compared with two common floorplanning approaches. 3D-DTC shows 24% increase in area efficiency as compared to single TSVs placement and 6% more area efficiency as compared to fixed-sized TSV islands.

iii) dynamic TSV clustering without KOZ sharing. The difference between case (ii) and (iii) is that in case (iii), when two single-TSVs merge to form a cluster, the spacing between TSVs equals to the sum of the KOZs in individual TSVs, whereas in case (ii) the spacing between adjacent TSVs in a cluster is fixed and equals to twice of TSV diameter as recommended by the ITRS[. Therefore, unlike case (ii), the TSV footprint in case (iii) remains unchanged with each clustering move.

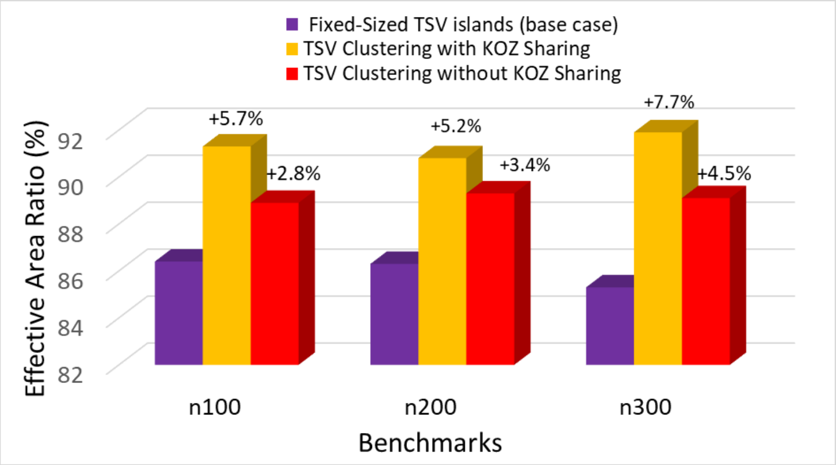


Figure 24. Comparison of area efficiency between three cases – (i) Fixed-sized TSV islands (ii) 3D-DTC with shared KOZ in TSV clusters (iii) 3D-DTC without sharing KOZ in TSV clusters. Results shown for case(ii) and (iii) are relative to base case (i).

Fig.24 compares the area efficiency of all three cases. Compared to the base case, the EAR improves by 6.2% in case (ii) and by 3.7% with case (iii) averaged for all benchmarks. It is notable that the improvement in case (iii) is solely due to the dynamic TSV clustering without any contribution of KOZ sharing. Based on the above results, the dynamic TSV clustering results in better packing efficiency than using TSV islands of fixed size, even without the influence of KOZ sharing amongst TSVs. This can be explained by packing amongst a wide spectrum of variable-sized TSV clusters, which potentially improves the utilization of whitespaces amongst the modules. This significantly raises the packing efficiency, as compared to using fixed-sized TSV islands. In addition, with the usage of fixed-sized TSV islands, one has to be carefully and selective of the appropriate number and capacity of islands for a design. This problem is eliminated with dynamic clustering approach, as it inherently optimizes the number, size and positions of TSV clusters to minimize packing area.

Table 8. Comparison of wirelength and number of TSVs with Lin et al [52] using wirelength-aware cost function *CF1.* Number of device layers = 4. TSV D = 3 µm, H = 30 µm, p = 6 µm

Benchmark	***Wirelength (mm)***			***No. of TSVs***		
	Lin et al [52]	3D-DTC	% diff	Lin et al [52]	3D-DTC	%diff
n100	131.793	119.4	-9.4	660	611	-7.2
n200	234.484	226.8	-3.6	1442	1281	-11.0
n300	323.975	301.7	-6.3	1443	1372	-4.9
			-6.4%	Avg		-7.1%

3.7.4. Wirelength-aware 3D Floorplanning with TSV Clustering

We compare the efficiency of our approach with Lin *et al* [52] in which, a four-phased TSV-aware 3D floorplanning is proposed with single stand-alone TSVs. First, modules are deterministically assigned to tiers in a layer assignment stage using a SA-based minimum-cut partitioning. A global distribution stage simultaneously distributes modules and single stand-alone TSVs to fixed placement regions in all tiers. The exact locations of TSVs are determined by a minimum cost maximum flow algorithm, optimizing wirelength and routability at the same. For fair comparison, we set the chip aspect ratio to 1 with 15% whitespace allocation. Table 8 shows results of our floorplanning using cost function CF1 which separately optimizes the total wirelength and number of assigned TSVs. The results are compared with hard module benchmark results in [52] considering four tiers. Our approach shows a significant 6.4% reduction in wirelength and about 7% reduction in TSVs averaged for all three benchmarks. As compared to placement of single TSVs in [52], the formation of a wide-variety of TSV cluster sizes enables better distribution of TSVs, by effectively utilizing the smaller spaces amongst the circuit blocks. This increases the availability of TSVs to a 3D net during assignment, thereby playing an important role in wirelength reduction.

Table 9. Comparison of proposed floorplanning (3D-DTC) with fixed-sized TSV islands approach [47] using delay-aware cost function – CF2 (($\alpha = 1.0$, $\gamma = 100$)

Benchmark	*Area (mm^2)*			*Total Interconnect Delay (ps)*		
	[47]	3D-DTC	%diff	[47]	3D -DTC	% diff
n100	0.263	0.237	-9.8	833	782	-6.1
n200	0.302	0.296	-4.6	1648	1526	-7.4
n300	0.481	0.473	-1.7	2492	2387	-4.2
Average			-5.4%			-5.9%

3.7.5. Delay-Aware 3D Floorplanning with TSV Clustering

Table 9 shows the results of delay aware-3D floorplanning with dynamic TSV clustering. The results are compared with floorplanning with fixed-sized TSV islands [47]. Compared to [47], the floorplan packing area improves by 5.4% averaged for all benchmarks, due to significant reduction in TSV area. The total interconnect delay reduces by 5.9% averaged for all three benchmarks. It is notable that reduction in total delay is not just due to reduced TSV footprint, but it is also attributed to the improved optimization of delay in individual 3D nets. In [47], the TSV RC delay contribution to its assigned wire was minimized by optimizing the number of TSVs assigned to each wire, based on length of the wire. Due to the quadratic dependency of delay on no. of TSVs (N_{TSV}), individual net delay was minimized. In addition, our nets-to-TSVs assignment also considers minimizing the TSV capacitance contribution (C_{TSV})to the assigned wire, by selecting the appropriate size of cluster.

3.7.6. Comparison of Runtime with Fixed-sized TSV Islands Approach

Fig.25 shows the runtime of our approach when compared with using three benchmarks. The runtimes are compared with the runtime obtained by floorplanning with fixed-sized TSV islands. Based on the three tested benchmarks, it is observed that the proposed approach increases the runtime by average 14%. This is due to significantly reduced solution space using islands of fixed sizes. Interestingly, with increase in size and complexity of the input benchmark from n100 to n300, the increase in runtime is larger (about 2.9x) for fixed-sized TSV islands, as compared to 3D-DTC whose runtime increases by 2.68x from n100 to n300. This can be possibly due to longer time for convergence required when the same fixed size and number of islands are used for all three benchmarks, irrespective of the circuit size.

The runtime can be reduced by certain measures to allow faster convergence. For e.g., it is observed that a cluster whose capacity '*m*' is a prime number then the only possible size of the resulting cluster is (1 x *m*). For a large capacity value, such a cluster will have a highly skewed aspect ratio, which may potentially degrade packing efficiency or cause a large obstacle for routing of signal nets. In the current version of the floorplanner, the selection of optimized size for a given cluster capacity is purely guided by the evaluation of floorplan packing area. There are no constraints applied to possible sizes during clustering moves. As a result of such a large solution space, the search for optimal cluster size incurs a longer runtime. This can be mitigated by restricting cluster capacities to composite integer values only. Although, this may potentially lead to faster convergence, further experiments are required to observe the impact of this measure on solution quality.

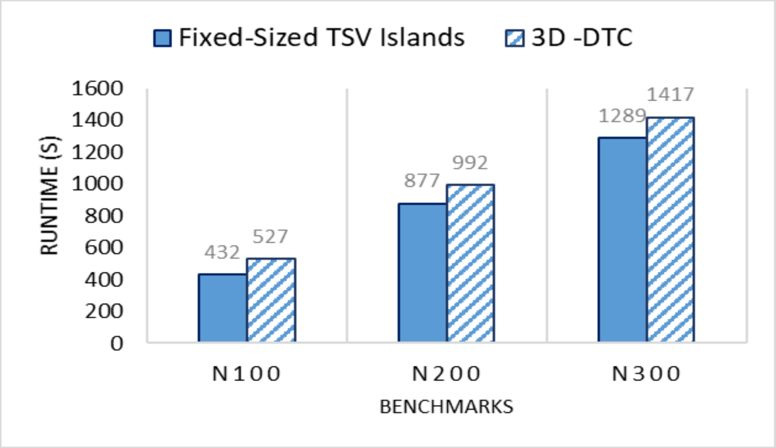


Figure 25. Runtime of our approach compared with floorplanning with fixed-sized TSV islands

3.7.7. Impact of Probability of Selection of Clustering Moves

We examine two test cases to analyze the impact of assigned probability of occurrence on TSV clustering. Fig.26(a)&(b) show the distribution of TSV cluster sizes on top three device layers for benchmark n100. Fig.26(a) shows case I where equal probability of occurrence (p=0.45) is assigned to merge and divide moves through all stages of floorplanning. Fig.26(b) shows case II, where the floorplanning is performed with the nonuniform probability for each move as per Table 5. In case I, the large probability of division in the initial stages, negates the impact of merges between clusters. Hence, a large population of small cluster sizes (1x1) and (1x2) are obtained which significantly increases the floorplan packing area. In case II, with the careful selection of assigned probability of moves, the packing area significantly improves by 17%. This is due to the formation of varied sizes of TSV clusters in the intermediate and final stages of floorplanning, which not only optimizes TSV area, but also improves packing efficiency.

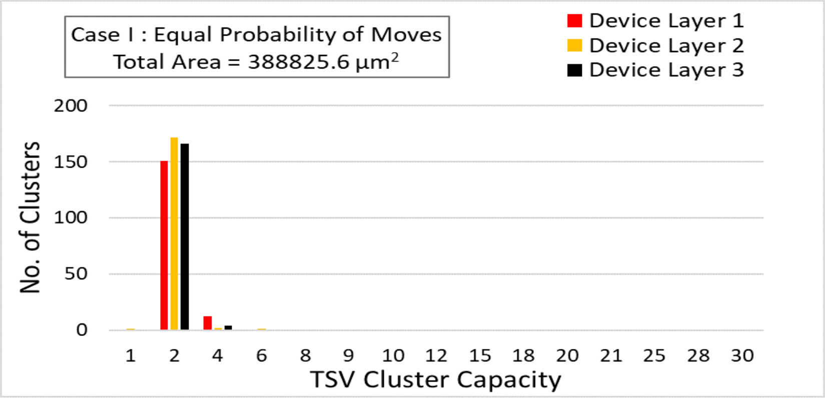


Figure 26(a). Cluster size distribution when merge and divide are assigned equal probabilities of selection through all three stages of floorplanning (p=0.45)

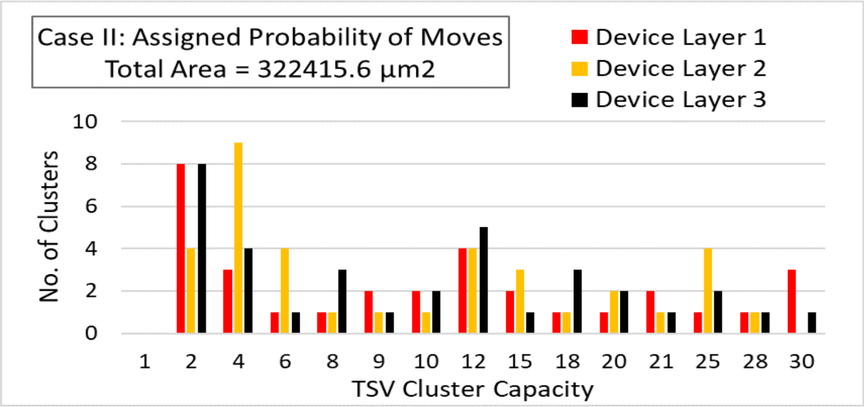


Figure 26(b). Cluster size distribution when merge and divide are assigned dynamically changing probabilities as per Table 5. (GSRC Benchmark: n100)

3.8. Chapter Conclusions

A new algorithm of dynamic TSV clustering during 3D floorplanning is proposed for improved optimization of interconnect performance. Our approach results in improved solution quality, as it dynamically optimizes the size and location of TSV clusters on the 3D layout. It also gives more realistic evaluation of the generated 3D floorplans. During TSV clustering, the direct minimization of TSV footprint through KOZ sharing plays a key role towards better optimization of both area and delay. However, our experiments reveal that even without the contribution of KOZ sharing, TSV clustering method will still be effective towards area minimization due to the increased efficiency of module-packing with a diverse range of cluster sizes. The availability of wide-ranging sizes of TSV clusters also allows better utilization of space among the circuit blocks, therefore, improving wirelength. With delay-aware cost function, our nets-to-TSVs assignment effectively minimizes the capacitance contributed by the assigned TSV to the net, by selecting the appropriate size of TSV cluster. We gain a deeper insight into the impact of strategic assignment of probabilities of occurrence for clustering moves. The key to faster convergence, is to base the selection of clustering moves not only on the assigned probability, but also on the solution quality at various stages of floorplanning. In this work, we have shown a bottoms-up clustering approach, such that we begin with smallest possible cluster size (1x1) and generate 3D layouts with clusters of variable sizes. However, the proposed approach is not restrictive in this context. It is also scalable to perform a top-down approach, starting with large-sized clusters and iteratively generate varying sized clusters within the optimization loop. For enabling such a top-down approach, more experimentation is required for judicious assignment of probabilities of occurrence for the clustering moves.

4. EARLY BUFFER COUNT ESTIMATION IN 3D FLOORPLANNING

Interconnect-driven timing optimization is a serious concern in building of high-performance 3D ICs, due to the potentially large signal and slew degradation induced by large-sized TSVs' capacitance. Hence, timing optimization is a necessary step at all stages of design, more so when TSVs lie in signal paths which are timing critical. The traditional method of buffer planning is still relevant for timing optimization in 3D ICs [75]. Buffer insertion has been traditionally used to linearize the quadratic dependence of delay on length of the wire [76]. While buffer insertion in traditional 2D ICs is primarily done in the later stages with available routed net topologies. However, similar strategies may not be adopted in 3D designs, due to the inherently large and complex design space of 3D ICs with inclusion of TSVs. 3D floorplanning is an important phase in the early design exploration of the 3D ICs, during which important metrics (area, delay, power) are evaluated and optimized. Therefore, incorporating buffer estimation during floorplanning has excellent potential for achieving better allocation of buffering resources during routing stage, accelerating timing convergence and design closure.

Early stage buffer estimation techniques are expected to play an even more important role for the success and viability of future 3D ICs . With shrinking of feature sizes by about 1.4x with each advancing node, the intrinsic delay of buffer is expected to decrease. However, due to the shrinking wire dimensions (1.4-1.5x per generation), the number of buffers is expected to increase dramatically due to wire delay [79]. The introduction of such large number of buffers can even introduce changes in the floorplan, degrading the quality of design and increasing costs. Therefore, inclusion of buffer estimation early during floorplanning will be even more crucial for future 3D designs, for lesser optimization loops post-routing and for achieving faster convergence.

This chapter addresses the problem of accurate evaluation of the number of buffers based on their estimated positions during 3D floorplanning. In early stages of design, we do not need to know the exact positions of buffers on the layout. Rather, the goal is to improve the accuracy and reliability of the estimation of buffer count, to generate better optimized 3D floorplans. The actual placement of buffers is possible in the later stages when precise information on routed topology of nets is available.

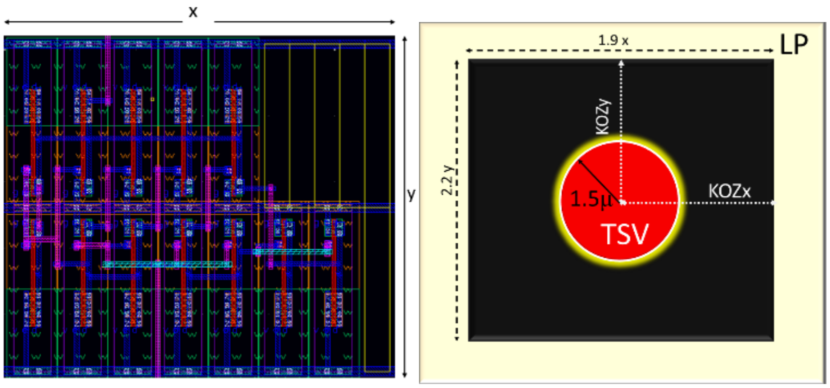


Figure 27. Comparison of area occupied between a buffer cell and a single -TSV cell as considered in our approach. (left) a single buffer cell (8x) at 45nm (right) top view of a 3um-diameter TSV cell.

4.1. Challenges of Early Buffer Estimation in 3D ICs

Unlike traditional 2D ICs, buffer insertion in TSV-based 3D ICs needs careful consideration of several additional constraints pertaining to the usage of TSVs. It is commonly believed that available 2D EDA tools can be slightly modified to be utilized for timing optimization in 3D ICs. However, such approaches result in limited solution quality as each die is handled separately, rather than acknowledging the whole 3D signal path containing TSVs. This implies that developing an effective buffering approach for timing optimization in 3D ICs has to be fundamentally different than 2D ICs.

Buffer estimation in 3D ICs is primarily influenced by three major factors: (i) finite area occupied by the TSVs (ii)TSV RC parasitics and (iii) position of TSV on the wire [75][78]. As shown in Fig.27, the area of a single (8x) buffer cell at 45 nm is almost 25% of the area of a single 3um-TSV cell. The finite area occupied by buffers introduces new design complexity, as buffers cannot interfere with the positions of TSVs or circuit blocks on the layout. Therefore, estimating the candidate locations of buffers simultaneously with nets-to-TSVs assignment phase can be a key factor in ensuring that the timing requirements of 3D nets are met without violating TSV locations, as compared to buffer insertion methods for routed net topologies. Secondly, in addition to the impact of TSVs' RC parasitics, the intrinsic delay of buffers contributes significantly to net delay and power therefore, requires appropriate characterization. This further implies that a fast and reliable model for delay estimation of buffered 3D nets is central to developing an efficient approach for buffer estimation in 3D ICs. Finally, the position and distribution of TSVs on the layout not just impacts the area and wirelength but also determines their capacitance contribution to the assigned net. To optimize the usage of buffers in individual 3D nets, it is also critical to consider the position of the assigned TSVs on the layout.

4.2. Previous Works

Recently, some prior works on buffer insertion in 3D-ICs have considered TSVs as obstacles along with functional blocks. [75]-[80]. Dong *et al* [74], proposed a buffer planning algorithm at the floorplanning stage. However, TSV parasitics was not considered therefore, limiting the accuracy of their estimated delay. Also, buffer delay was assumed to be constant, irrespective of RC load. In [75], He *et al*. considered redistribution of available white-spaces in floorplan for simultaneous insertion of buffers with TSVs. Although the redistribution approach improves the TSV allocation and buffer insertion rate, the solution is limited and degrades the total wirelength and overall packing. Also, they ignore the TSV RC delay impact on net delay. Kim *et al*. [78] first considered the significant contribution of TSV RC parasitics towards delay of buffered 3D interconnects. However, they used a fixed interval between consecutive buffers (buffer insertion length) for both 2D and 3D nets which is not realistic. The actual TSV locations along the wire was not considered to estimate number of buffers for a net. The above factors severely limit the accuracy of the estimated interconnect delay and power. In [76] Lee *et al*. proposed fast delay estimation techniques for buffered 3D interconnects. They considered the effect of TSV parasitics and their positions to find optimum location of buffers in a 3D net. However, the approach was based on a fixed floorplan with assigned TSV positions. They ignore the significant impact of coupling between adjacent TSVs on TSV capacitance. Individual nets of fixed lengths were used to verify the approach rather than standard benchmarks. In addition, none of the prior works have incorporated the problem of optimizing TSV position and distribution during floorplanning, which has critical impact on the estimated number of buffers and their positions in 3D interconnects. In [77], the TSVs' positions and RC delay are considered to derive the optimal distance between consecutive buffers. However, the delay of buffered segments containing TSV is not characterized accurately, requiring additional buffers next to TSVs in order to prevent signal deterioration.

In this chapter we present the following contributions:

- A method for fast and accurate estimation of the number and positions of buffers simultaneously with nets-to-TSVs assignment during 3D IC floorplanning is developed. This allows the consideration of three critical factors which impact interconnect delay and power - i) TSV area ii) TSV RC parasitic delay contributed to the net and iii) positions of assigned TSVs along the net. Compared to past approaches, we accurately characterize the delay of a buffered 3D net for estimation of optimal distance between consecutive buffers.
- We propose a method to find the optimal positions of buffers before TSVs. This helps to minimize the excessive use of buffers around TSVs. An in-depth analysis of the potential impact of sub-45 nm technologies and nano-scale TSVs on buffer estimation in future 3D ICs is presented.

Table 10. List of notations used in proposed methodology of buffer estimation in 3D ICs

Notation	***Definition***
L	Estimated wirelength of a 3D net
BIL	Buffer insertion length
N_{TSV}	Number of TSVs in a 3D net.
N_{buf}	Number of buffers estimated for a 3D net
N_{2D}	Number of 2D segments in a net
N	Total number of buffered segments ($N_{TSV} + N_{2D}$)
L_{3D}	Total length of all 3D segments in a 3D net
L_{2D}	Total length of all 2D segments in a 3D net
L_{rem}	Known interval between TSV and current buffer
k	Distance between projected position to optimal position of buffer in a 3D segment

4.3. Estimation of Buffer Insertion Length in 3D Interconnects

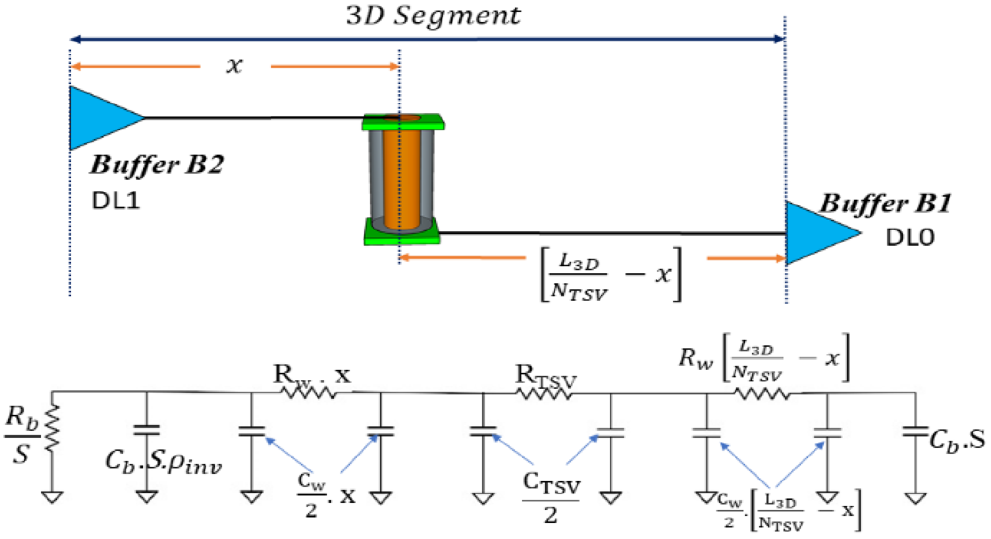
A) Definitions - The following important terminologies are defined in our approach as shown in Table 10. i) *Buffer insertion length (BIL)* is the minimum required distance between consecutive buffers below which, placing an optimally sized buffer reduces the delay of the corresponding un-buffered wire. *(ii)* A *3D net* is a net which spans across 'm' consecutive device layers and requires (m-1) TSVs. A *2D wire* is a two-pin net with its terminal pins (driver and sink) located on the same device layer. *(iii)* A *3D segment* is a single wire segment, containing a TSV and one buffer. *(iv)* A *2D segment* is a single wire segment with a buffer and is contained on one device layer. From Table 10, the number of 3D segments (N_{3D}) and the number of buffers in these segments is equal to the number of TSVs (N_{TSV}) in a 3D net. The number of buffers in 2D segments is equal to (N_{2D} -1). Therefore, the total number of buffers (N_{buf}) in a 3D buffered net equals to (N_{TSV} + N_{2D} - 1*)*.

B) Delay Estimation for Buffered 3D Interconnects

We focus on point-to-point 3D nets to enable fast buffer count estimation. A fast and reliable estimation is necessary to reduce the computational cost of probabilistic heuristics used for generating 3D floorplan solutions. We use a quick and fairly accurate Elmore delay-based estimation, that is suitable for the early design stage. In this stage, we usually cannot determine signal slews on nets, hence ignored. In an optimally buffered 3D net, the total length 'L' consists of both 2D segments and 3D segments. as represented by (1):

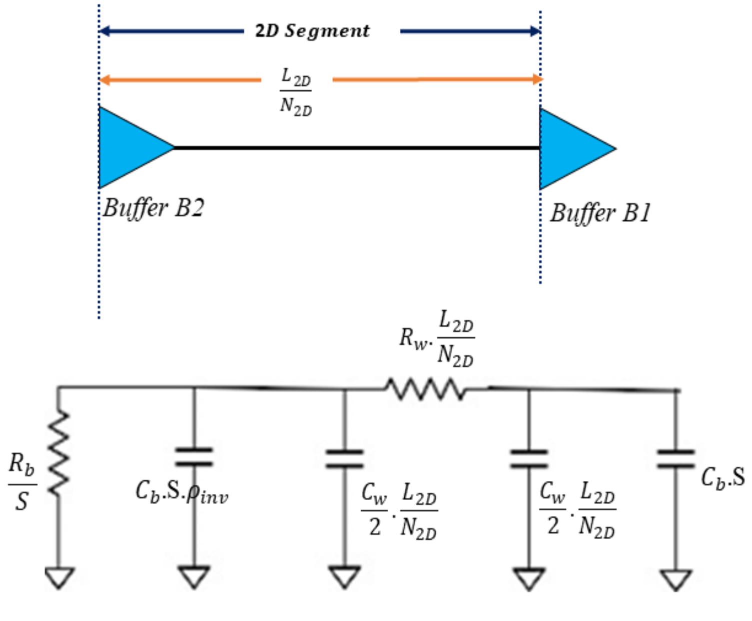
$$L = L_{2D} + \ L_{3D} \tag{1}$$

Fig.28(a) shows a typical 3D segment spanning across consecutive device layers DL1 and DL0 and containing one TSV. Buffer B2 is located on DL1 at a distance 'x' from the TSV. The



3D Segment
x
Buffer B2
DL1
Buffer B1
DL0
$\left[\frac{L_{3D}}{N_{TSV}} - x\right]$
$R_w . x$
R_{TSV}
$R_w \left[\frac{L_{3D}}{N_{TSV}} - x\right]$
$\frac{R_b}{S}$
$C_b.S.\rho_{inv}$
$\frac{C_w}{2}. x$
$\frac{C_{TSV}}{2}$
$\frac{C_w}{2}.\left[\frac{L_{3D}}{N_{TSV}} - x\right]$
$C_b.S$

Figure 28(a). Comparison of area occupied between a buffer cell and a single -TSV cell as considered in our approach. (left) a single buffer cell (8x) at 45nm (right) top view of a 3um-diameter TSV cell..



2D Segment
$\frac{L_{2D}}{N_{2D}}$
Buffer B2
Buffer B1
$R_w.\frac{L_{2D}}{N_{2D}}$
$\frac{R_b}{S}$
$C_b.S.\rho_{inv}$
$\frac{C_w}{2}.\frac{L_{2D}}{N_{2D}}$
$\frac{C_w}{2}.\frac{L_{2D}}{N_{2D}}$
$C_b.S$

Figure 28(b). Elmore delay RC equivalent circuit of 2D segment

delay of a 3D segment is contributed by the driving buffer B2, the TSV and associated wire segments. From the distributed RC model, the delay of 3D segment 'D_{3D}' is formulated as:

$$D_{3D} = \frac{R_b}{S}.\left[C_w.x + C_b.S(1+\rho_{inv}) + C_{TSV} + C_w\left(\frac{L_{3D}}{N_{TSV}} - x\right)\right] + R_w.x\left[\frac{C_w}{2}.x + C_{TSV} + C_w.\left(\frac{L_{3D}}{N_{TSV}} - x\right) + C_b.S\right] + R_{TSV}.\left[\frac{C_{TSV}}{2} + C_w.\left(\frac{L_{3D}}{N_{TSV}} - x\right) + C_b.S\right] + R_w.\left(\frac{L_{3D}}{N_{TSV}} - x\right)\left[\frac{C_w}{2}.\left(\frac{L_{3D}}{N_{TSV}} - x\right) + C_b.S\right] \tag{2}$$

R_{TSV} and C_{TSV} represent the TSV's resistance and capacitance respectively. The TSV resistance has two components – (i) material resistance and (ii) contact resistance between TSV and landing pads at both ends [78]. TSV capacitance is computed using analytical models from [55]. The inductance of TSV is ignored as it is not dominant at a few GHz signal speed [78]. R_b and C_b represent the buffer output resistance and input capacitance respectively. For buffers, we assume a linear RC delay model with lumped load capacitance. The driver resistance is assumed equal to the buffer output resistance. The load capacitance of the sink is assumed identical to the buffer input capacitance. 'S' represents the buffer size expressed in terms of the number of times unit buffer size. 'R_w' and 'C_w' represent the unit wire resistance and capacitance respectively. Wire capacitance is computed assuming uniform wire density on all device layers. 'ρ_{inv}' represents the parasitic capacitance factor taken as 0.5 [8]. First, we find the optimal position 'x' of the driving buffer, for which delay of a 3D segment is minimized. Hence, differentiating (2) by 'x' we get, $(dD_{3D} / dx) = R_w.C_{TSV} - R_{TSV}.C_w$. Since, $R_w.C_{TSV} \gg R_{TSV}.C_w$, $(dD_{3D} / dx) > 0$. Thus, D_{3D} will be minimum when $x = 0$, i.e., buffer B2 is placed right before the TSV, after the required radius of KOZ. Hence, substituting $x = 0$ in (2) we get:

$$D_{3D} = \frac{R_b}{S} . \left[C_b.S(1+\rho_{inv}) + C_{TSV} + C_w.\frac{L_{3D}}{N_{TSV}}\right] + R_{TSV} . \left[\frac{C_{TSV}}{2} + C_w.\frac{L_{3D}}{N_{TSV}} + C_b.S\right] + R_w.\frac{L_{3D}}{N_{TSV}}.\left[\frac{C_w}{2}.\frac{L_{3D}}{N_{TSV}} + C_b.S\right] \quad (3)$$

From Fig. 28(b), the delay of a 2D segment 'D_{2D}' is given by (4) as:

$$D_{2D} = \frac{R_b}{S}\left(C_b.S(1+\rho_{inv}) + C_w.\frac{L_{2D}}{N_{2D}}\right) + R_w.\frac{L_{2D}}{N_{2D}}\left(\frac{C_w}{2}.\frac{L_{2D}}{N_{2D}} + C_b.S\right) \quad (4)$$

The required minimum BIL equals to the length of a 2D segment, i.e., $\left(\frac{L_{2D}}{N_{2D}}\right)$. It is notable that due to the difference between the delay of 2D segments and 3D segments, the length of a 2D segment differs from a 3D segment. For e.g., at 45nm technology, a 3D segment with a 3µm-diameter TSV is about 5x shorter than a 2D segment. Past approaches [7][8] have inaccurately considered the same BIL interval to represent both 2D and 3D segments. We distinguish between delay of 2D and 3D segments, to estimate the optimal BIL as follows. The total delay of a buffered 3D net 't_{pd}' is calculated as:

$$t_{pd} = N_{2D} * D_{2D} + N_{TSV} * D_{3D} \quad (5)$$

The minimum required BIL is found by differentiating (5) w.r.t 'N_{2D}' and 'S'. The resulting expression is set to zero (the minimum values of the argument) to solve for $\frac{L_{2D}}{N_{2D}}$, assuming the same size 'S' for all buffers during floorplanning.

$$\frac{L_{2D}}{N_{2D}} = \frac{\sqrt{[(R_{TSV}.C_w)^2 + 2.R_w.C_w[R_b.C_b(1+\rho_{inv}) + N_{TSV}.R_{TSV}.C_b]]}}{R_w * C_w} - \frac{R_{TSV}}{R_w} \quad (6)$$

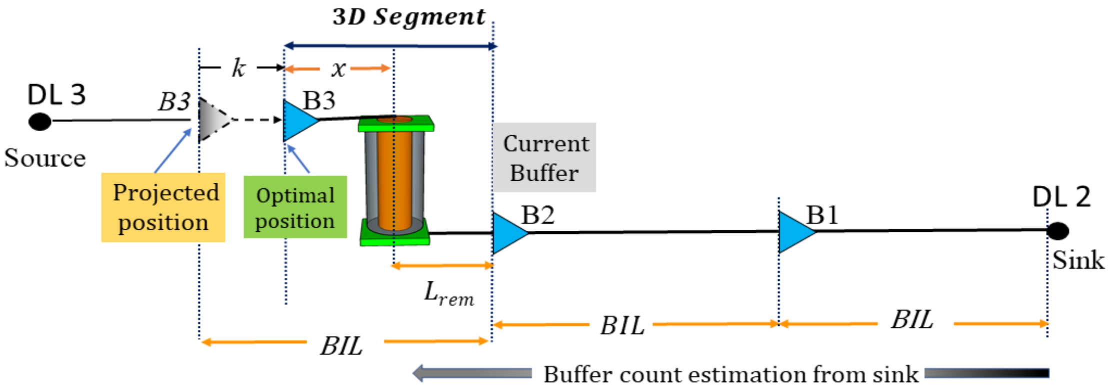


Figure 29. The estimated BIL is used to find candidate locations of buffers in a two-pin 3D net by traversing the net from sink-to-source

It can be observed that, for a 2D net (when $N_{TSV} = 0$ and $R_{TSV} = 0$), we obtain the standard expression for optimal BIL of a 2D net of length 'L' with 'N' repeated segments given by (7):

$$\frac{L}{N} = \sqrt{\frac{2R_b * C_b(1+\rho_{inv})}{R_w * C_w}} \tag{7}$$

Factors influencing BIL – For a given buffer size and wire dimensions, the optimal BIL is independent of wirelength and TSV capacitance, while it is influenced by TSV resistance. The BIL increases with number of TSVs assigned to the net. This is explained by the increase in TSV delay contribution relative to wire delay, represented by the parameter *TSV equivalent wirelength* [4]. On the other hand, for number of TSVs, increasing the wire RC delay has the opposite effect. In this case, the BIL decreases, as total load driven by buffer increases.

C. Steps of Buffer Count Estimation in a 3D Interconnect - Fig.29 shows an example of a two-pin 3D net with one TSV, spanning across consecutive device layers DL3 and DL2. In case of multi-pin 3D nets, the buffers' positions are estimated after decomposing the net into two-pin nets. We perform a bottom-up traversal of the net from sink to source. The candidate location of buffer *B1* is found at BIL interval from sink. Similarly, the next buffer *B2* is located at BIL distance from current buffer *B1*. If a TSV is found in the path of traversal such that the remaining distance between current buffer *B2* and the TSV is less than BIL, then that interval is labeled as 'L_{rem}'.

Finding Optimal Buffer Position in a 3D segment – Fig.29 shows the projected position of buffer *B3* at BIL interval from *B2*. From the projected position, buffer *B3* has to be moved by distance *'k'*, to reach its optimal position '*x*' from the TSV. A buffer is optimally placed, if it drives the total output capacitive load of the 3D segment without incurring signal degradation across TSV. Hence, at the optimal position '*x*', the output capacitance load for *B3* includes- the TSV capacitance (C_{TSV}), capacitance of wire segments (C_w*x+ C_w* L_{rem}) and capacitance of the downstream buffer *B2* (C_b). To find '*k*', we formulate the path delay of the *projected* 3D segment 'D_{proj_3D}' as:

$$D_{proj\_3D} = R_{\ b}[C_w\,(k+x) + \ C_b(1+\rho_{inv}) + C_{TSV} + C_w . L_{rem}]$$

$$R_w\,(k+x)\left[\frac{C_w}{2}(k+x) + C_{TSV} \ + C_w . L_{rem} + \ C_b\right] +$$

$$R_{TSV}\left[\frac{C_{TSV}}{2} + \ C_w . L_{rem} + \ C_b\right] + R_w . L_{rem}\left[\frac{C_w}{2} . L_{rem} + \ C_b\right] \tag{8}$$

We set $x = 0$, in (8), as we consider the optimal position of driving buffer in front of TSV, for which 3D segment delay is best minimized, as described earlier in section III (B). The resulting expression is differentiated by '*k*' and the result is set to zero, to find the minimum '*k*' as shown in (9):

$$k = \frac{C_{TSV} + C_w \,.\, L_{rem} + C_b}{C_w} \qquad (9)$$

The benefit of finding the optimal position of the driving buffer before a TSV is that there is no longer the need to insert additional buffers before and/or after TSVs, as done in prior methods [77][78]. If 'L_{rem}' is large, buffer shifts more towards the TSV. In some 3D nets requiring more than one TSVs, the optimal buffer position on the lower device layer may fall before the coordinates of the TSV on the upper device layer. In this scenario, the buffer is added right after the TSV on the upper layer. The candidate position of the next buffer is then estimated based on the position of the added buffer.

D. Dynamic Power of 3D Buffered Interconnects - The dynamic power dissipation of a 3D buffered interconnect ($P_{\text{interconnect}}$) is evaluated according to (10). α is the switching activity taken as 0.3, V_{DD} is the supply voltage and 'f' is the operating frequency.

$$P_{\text{interconnect}} = \alpha * f * V_{DD}^{2} * (Cw * L + C_{TSV} * N_{TSV} + N_{buf} * C_b) \qquad (10)$$

4.4. Influence of TSV Cluster Size on Buffer Estimation

In our approach, TSVs are arranged in clusters (regular arrays) whose sizes vary dynamically through iterations, resulting in effective optimization of TSV footprint and TSV capacitance. Each cluster area includes a TSV pitch between adjacent TSVs and required keep-out zone around the cluster periphery [81]. We assume that capacitance (C_{TSV}) of each TSV in a cluster is same and equals to the capacitance of TSV subjected to maximum coupling from its neighboring TSVs. This TSV is the worst-case capacitance (highlighted yellow). Fig. 30 shows the merging operation of two small sized clusters A and B, resulting in a single large cluster C. It can be observed that due to change in the number of TSVs surrounding the worst-case capacitance TSV in A and B, the TSV capacitance (C_{TSV}) of resulting cluster C increases by 63% and 29% respectively. Therefore, during our buffer insertion, the size and aspect ratio of a TSV cluster selected for assignment to the net matters, as it determines the TSV capacitance contribution to both delay and power of the net.

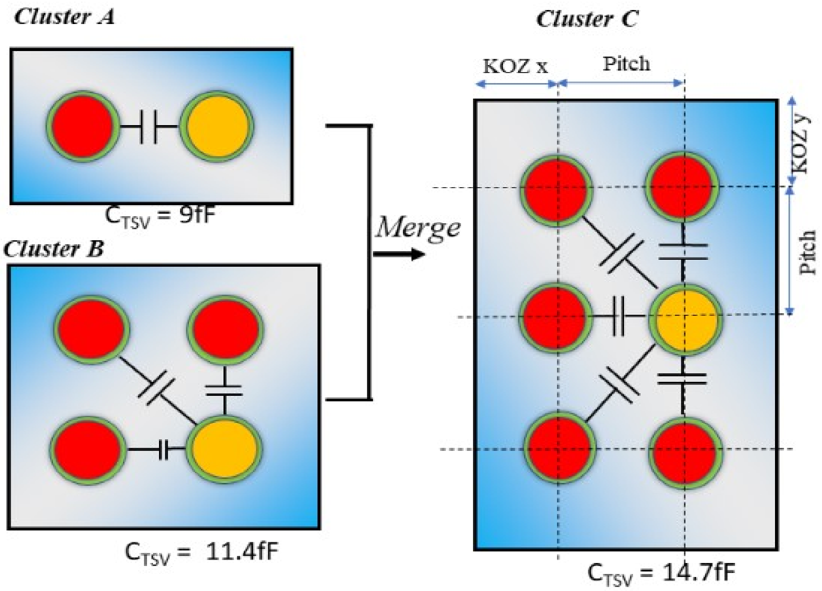


Figure 30. Effect of TSV clustering on TSV-capacitance

4.5. General Flow of 3D Floorplanning with Buffer Estimation

The proposed buffer insertion method is incorporated in our 3D floorplanner, which is based on dynamic variation of TSV cluster sizes. During the optimization stages, dynamic assignment of nets to TSV clusters is employed only after the four-tier floorplan area reaches a desired minimum percentage. For estimation of wirelength of a 3D net, we use an accurate net-splitting method, where each 3D net is split into subnets based on pin location on device layers and assigned TSVs. The total wire length is the summation of HPWL of all subnets. Depending on the length of the wire and position of assigned TSVs, buffer locations are estimated as described in Section 4.4. The generated output is a floorplan with optimized area and total delay. The parameters - number of buffers and dynamic interconnect power are evaluated on the final floorplan.

Table 11. Technology Specifications used in our experiments

Device Technology	*45 nm*	*32 nm*	*22 nm*
Metal Width (nm)	140	112.5	80
Metal Thickness (nm)	252	204	160
Metal Spacing (nm)	140	112.5	80
Unit wire resistance (mΩ/μm)	487.5	749.5	1343.8
Unit wire capacitance (fF/ μm)	0.171	0.129	0.096
Supply Voltage (V)	1	0.9	0.8
Operating Frequency (GHz)	2	2.3	2.6
Buffer Intrinsic Delay (ps)	1.99	1.41	0.99
Buffer Input Capacitance (fF)	6.65	4.71	3.31

4.6. Experimental Results

We implemented our 3D floorplanning algorithm with buffer estimation in C++/STL. The experiments were performed on a 4xDual Core Sun SPARC IV CPUs at 1.35 GHz and total 32 GB RAM. We assume via-middle TSV technology with face-to-back (F2B) die-stacking, as it allows stacking of more than two device layers [4]. With the F2B stacking, a 3D net spanning between consecutive device layers will require TSV allocation only on the upper device layer. During floorplanning, all buffers are assumed to be of the same size. For experiments, we use expanded Gigascale Systems Research Center (GSRC) benchmarks as shown in Table 11. The original GSRC benchmarks have been expanded, primarily for comparison of our approach with previous works [77][78]. Secondly, as compared to smaller standard GSRC benchmarks, the expanded benchmarks enable us to simulate more complex circuits with longer interconnects, requiring more use of buffers.

Table 12. Modified GSRC benchmarks as used in our approach.

Test Case	Block Size	# Nets	# TSVs
n100	100x	885	924
n200	100x	2136	2032
n300	100x	2914	2825

4.6.1. Comparison with Previous Approaches [77][78]

Fig. 31 shows buffer count on four device layers obtained from floorplanning results of n200 benchmark circuit. Our results are compared with previous works [77][78]. In [78], a fixed buffer insertion length (350 μm) was used irrespective of 2D or 3D nets. We compare with their best case -BIS1 scheme, in which buffer is always placed right in front of a TSV. In [77], the BIL was computed by considering the delay of 2D segments and 3D segments as equal. Due to this inaccuracy, additional buffers were placed around each TSV to prevent non-existing signal degradation. For fair comparison, we assume RTSV to be equal to contact resistance of 40 Ω .

From Fig.31, the proposed approach results in average 22% reduction in buffer count for all device layers, as compared to fixed-BIL approach in [78]. As compared to variable-BIL approach in [77], the proposed method reduces buffer count by 7% averaged for all device layers. It is observed that the intermediate device layers *DL1* and *DL2* incur more TSV usage, compared to top *DL3* and bottom *DL0* device layers due to higher occurrence of 3D nets traversing through these layers. Our approach effectively reduces the buffer count in the intermediate layers by average 4%, as compared to [77]. The reduction in buffer estimates in the intermediate device layers can significantly ease the allocation of routing resources.

Table 13 shows the estimated buffer insertion length (BIL) obtained using proposed approach in (7), as compared to the estimated BIL in [77]. For purpose of comparison, we consider four buffer sizes and four types of nets in a 3D design, considering a maximum of 4 active layers- i) net with no TSVs ii) net with 1-TSV iii) net with 2-TSVs and iv) net with 3-TSVs. The estimated BIL in [77] is larger than the proposed approach by average (5%) for all buffer sizes. It is also observed that, even with a larger BIL in [77], their approach leads to overestimation of buffers in individual nets. This is because, buffer planning around TSVs did not consider the actual delay contribution of wire in the segment, to find an optimal position for buffer w.r.t TSV. Also, the length of TSV segment is assumed equal to BIL, which is not realistic. Consequently, the delay of TSV segment exceeded the delay of non-TSV segment due to significantly larger TSV delay contribution. In order to compensate for the signal degradation across TSV, an additional buffer was required at the beginning or end of TSV, depending on the position of TSV. This results in increased usage of buffers per net.

Table 14 compares our floorplanning results - buffer count, total interconnect delay and total interconnect power, with [77] and [78]. With the proposed method, the total number of buffers estimated for all benchmarks reduces by significant 21% as compared to [78] while interconnect delay and interconnect dynamic power reduce by average 6% and 8% respectively. Compared to [77], the number of buffers' estimate reduces by average 5.6% for all benchmarks. Interconnect delay reduces by average 5.7% and interconnect dynamic

Table 13: Estimated BIL using proposed approach compared with BIL in [77] for four buffer sizes. Metal Spec : 45 nm TSV Dia = 3µm, TSV Aspect Ratio = 10

Buffer Sizes	**Buffer Insertion Length (BIL) (µm)**									
	2D Net	**Net with 1-TSV**			**Net with 2-TSVs**			**Net with 3-TSVs**		
		BIL [77]	*Proposed*	*% diff*	*BIL [77]*	*Proposed*	*% diff*	*BIL [77]*	*Proposed*	*% diff*
4x	232.07	239.81	221.11	-7.8	243.01	228.71	-5.8	245.47	237.13	-3.3
8x	229.16	239.96	222.15	-7.4	244.44	228.96	-6.3	247.87	238.64	-3.7
16x	221.37	236.12	227.34	-3.8	242.23	233.46	-3.6	246.92	242.88	-3.2
32x	220.82	241.64	231.19	-4.3	250.27	240.14	-4.0	255.32	249.41	-2.3
Avg				-5.8			-4.9			-3.8

Table 14: Comparison of number of buffers, interconnect delay and total power dissipation with [77] and [78]. TSV Specifications: Dia = 3 µm, TSV Pitch = 6 µm, KOZ = 4.6 µm. S = BUFx_8

Modified GSRC Circuit	**Total Number of Buffers**			**Total Interconnect Delay (ns)**			**Total Interconnect Power (mW)**		
	Fixed BIL [78]	Variable BIL [77]	Our Approach	Fixed BIL [78]	Variable BIL [77]	Our Approach	Fixed BIL [78]	Variable BIL [77]	Our Approach
N100	6740	6557	6261	118.2	118.1	112.3	202	197.9	181.6
N200	21896	20696	19337	364.8	362.5	346.7	653.4	622.3	601.9
N300	36293	34319	32389	651.2	649.3	607.5	1082.8	1031.5	1004.4
Avg.	1.0	0.95	0.89	1.0	0.99	0.94	1.0	0.96	0.92

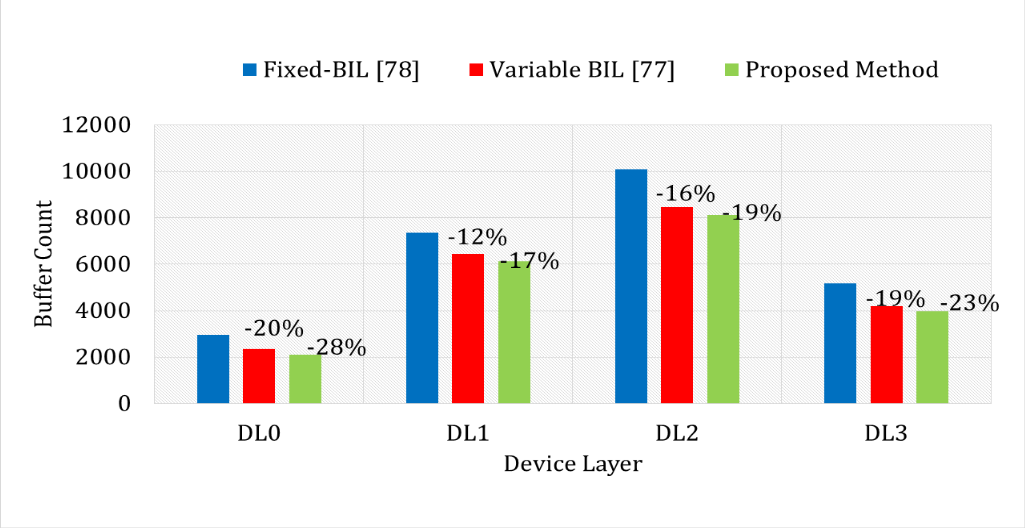


Figure 31. Comparison of buffer count on individual device layers (GSRC benchmark: n200). Data labels indicate percentage reduction w.r.t base case [78]

power by 4%. The reduction in interconnect power can mitigate temperature and power density issues which are major concerns in 3D ICs.

4.6.2. Impact of Process Technology on Buffer Insertion Length

The BIL is a function of both TSV resistance (R_{TSV}) and number of TSVs (N_{TSV}) in a 3D net (6). However, the extent of their impact on BIL will be critically influenced by the choice of technology node used for 3D IC fabrication. To demonstrate this, we consider three future nano-CMOS technologies, that are yet to be realized for 3D ICs – 45 nm, 32 nm and 22 nm [71]. BIL is estimated for TSV diameter of 3 μm with three types of 3D nets- 1-TSV net, 2-TSVs net and 3-TSVs net. The appropriate TSV contact resistance values at each node is taken from [107] assuming Cu as the TSV material. The wire RC delay at each node is computed using specifications in Table 11.

At each node, the buffer intrinsic delay ($R_b.C_b$) is scaled as shown in Table 11, using appropriate scaling coefficients from [108]. Fig. 32 shows, as technology scales from 45 nm to 22 nm, the BIL decreases significantly by 70% for 1-TSV nets, 67% for 2-TSV nets and 65% for 3-TSV nets. This reduction is due to a large 54% increase in wire RC delay combined with 49% reduction in buffer intrinsic delay.

The reduction in BIL with technology is comparatively higher for a 1-TSV net than a 3-TSV net. This phenomenon can be attributed to the ratio of R_{TSV}/Rw, which is 3x higher for a 3-TSV net than a 1-TSV net. This results in a smaller reduction in BIL in 3-TSV nets. As the number of TSVs increases in a net, BIL increases by 9% at 45 nm, 18% at 32 nm and 21% at 22 nm node. A key takeaway is that the relatively large increase in BIL at 22nm is due to the higher R_{TSV}/R_W ratio at 22 nm than 45 nm. This is caused by a large increment in TSV resistance at 22 nm [5].

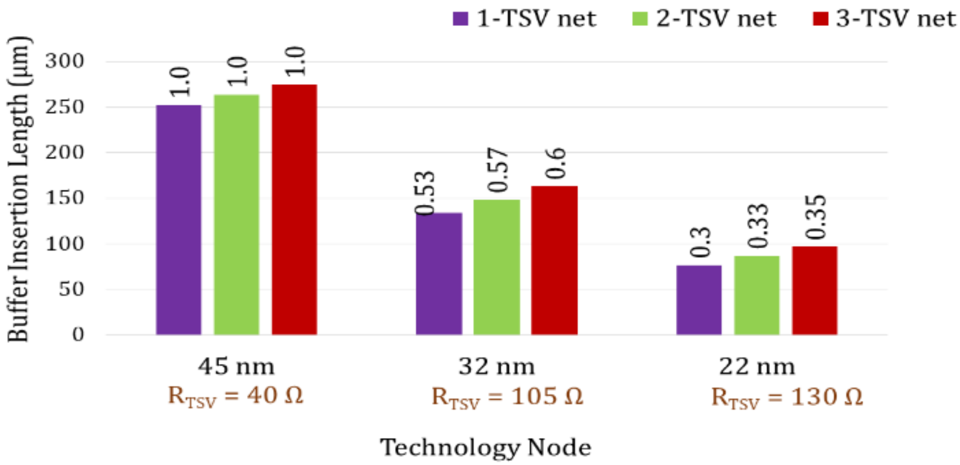


Figure 32. Variation of BIL with technology scaling. Data labels indicate BIL values normalized with respect to 45 nm node for each type of 3D net instance. (TSV dimensions: diameter = 3µm, aspect ratio =10).

4.6.3. Impact of Nano-scale TSVs on Buffer Insertion length

We compare the potential impact of nano-scale TSVs and micron-sized TSVs on optimal buffer position in a 3D segment (please refer Fig.29). In Fig.33, the variation of distance ‘k’ for two TSV diameters – 3 µm and 0.5 µm, at three future nanotechnologies - 45 nm, 32 nm, 22 nm is shown. For ‘Lrem’, a nominal value of 50 µm is assumed in each case. It is observed that TSV downsizing from 3 µm to 0.5 µm significantly reduces the distance ‘k’ by 48% at 45 nm, 56% at 32 nm and 65% at 22 nm. This is caused by the large reduction in TSV capacitance due to reduced parasitic coupling of TSVs with overhead wires and other neighboring TSVs within the same TSV island [73].

As delay of a 3D segment reduces with TSV downsizing, the optimal position of the buffer moves away from the TSV. This shifts the positions of subsequent buffers in the rest of the wire, potentially reducing the number of buffers in the net. For e.g., let us consider a 3D net of length 1050µm containing a 3µm-diameter TSV. We assume a 500 µm wire segment on the upper device layer and a 550 µm wire segment on the lower device layer. With estimated

BIL close to 250 µm at 45 nm technology, four buffers will be required, whereas for the same net with 0.5µm-TSV, only three buffers will be required.

When migrating from 45 nm to 22 nm, distance 'k' significantly increases by 56% for 3µm-diameter TSV. However, with nano-scale TSV, distance 'k' shows only a marginal increase of 7% at 22 nm, due to the combined impact of smaller TSV capacitance and increased wire capacitance, as compared to 45 nm [73]. This shows the potential of nano-scale TSVs in curbing the rate of increasing buffers in individual nets with technology, reducing both power and delay at lower nodes.

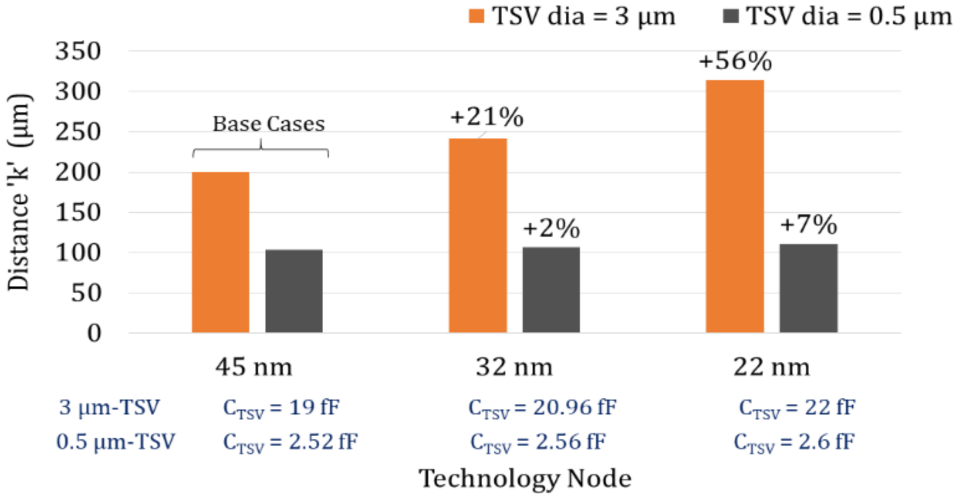


Figure 33. Variation of '*k*' with TSV downsizing and technology. The value '*k*' represents the distance by which buffer moves to its optimal position (Fig.29). Data labels show percentage increase in '*k*' w.r.t the base cases at 45 nm.

4.6.4. Impact of TSV downscaling on Performance in Buffered 3D ICs

In order to investigate the impact of nano-scale TSVs on buffers, we compare the floorplanning results of buffer count, total delay and power between two different TSV sizes – 3 µ and 0.5µm respectively as shown in Table 15. The aspect ratio of both TSV sizes is 10 as recommended by ITRS [4]. Due to variation in TSV width, the required KOZ dimensions are also incorporated as per analytical model in [80]. We consider an average of 25 floorplanning runs to generate all results. For each parameter, the average of all benchmarks is normalized to average of 3µm-TSV. With TSV downsizing from 3 µm to 0.5 µm, buffer count reduces by 26 % . This is due to a significant (81%) decrease in TSV capacitance with reduction TSV width to nanometer level, which decreases the delay of 3D segments. Simultaneously, with TSV downsizing, the increase in TSV resistance lengthens the BIL, thereby reducing the average buffer count per net. The decrease in buffer count reduces interconnect delay reduces by average 5% for all benchmarks. As compared to 3µm TSVs, the use of nanoscale TSVs, reduces dynamic power by average 8% for all benchmarks, due to reduction in both buffer count per net and TSV capacitance.

Table 15. Comparison of buffer count, total delay and total dynamic power for two different TSV sizes. Average of results for all benchmarks is normalized to TSV diameter 3 µ.

Circuit	**TSV Dimensions and Specifications**					
	TSV D = 3µ, H = 30 µ, PITCH = 6 µ, KOZ = 4.6 µ			TSV D = 0.5µ, H = 5 µ, PITCH = 1.0 µ, KOZ = 1.0 µ		
	Buffer Count	*Total Delay (ns)*	*Total Power (mW)*	*Buffer Count*	*Total Delay (ns)*	*Total Power (mW)*
N100-exp	6271	112.34	181.6	4812	109.31	173.2
N200_exp	19437	341.71	621.15	11653	322.71	603.31
N300_exp	32389	632.47	1004.4	26341	618.4	887.5
Normalized to 3 um	1.0	1.0	1.0	0.74	0.95	0.92

4.6.5. Impact of TSV Cluster Size on Optimal Buffer Location

Fig. 34 shows the impact of variation in TSV capacitance due to TSV cluster sizes, on the distance by which a buffer before TSV is moved from its projected to its optimal location. This interval is shown as '*k*' in Fig. 29. The horizontal axis represents the TSV cluster size (number of TSVs within the cluster) and the vertical axis represents interval '*k*'.

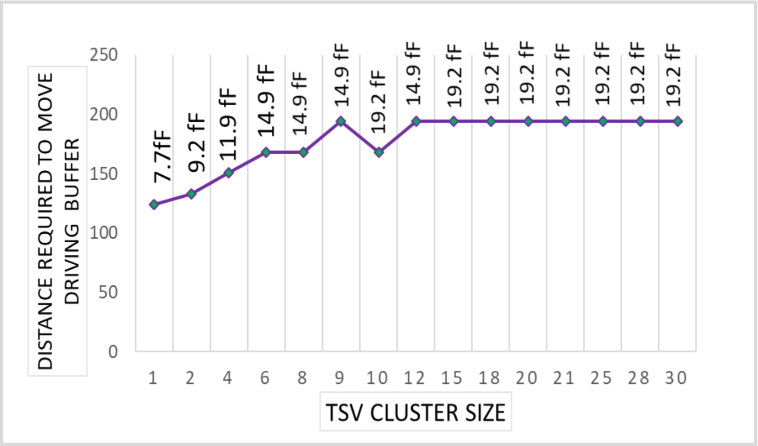


Figure 34. Impact of TSV cluster size on the distance required to move driving buffer to optimal position before a TSV.

The TSV capacitance value corresponding to each cluster size is mentioned as labels of data points. It can be observed, with increase in TSV cluster size from 1 to 30, TSV capacitance contribution increases by 18% due to increase in coupling with neighboring TSVs. This significantly increases the path delay of a 3D segment, causing the driving buffer to move closer to TSV (i.e., distance '*k*' increases). Hence, the delay of a 3D segment is not just impacted by the sizes of buffers, TSV and wire segments, but also depends on the size of the TSV cluster in which the assigned TSV is located.

4.7. Chapter Conclusions

A fast and accurate method for buffer count estimation is developed that notably improves the accuracy of evaluating delay and power - two vital measures of goodness of a 3D floorplan. Accurate characterization of delay of 3D segments is important to minimize excessive use of buffers in two ways – (i) by obtaining a more reliable estimate of buffer insertion length and (ii) by finding the optimal position of a buffer in a 3D segment, the need for extra buffers around the TSV is reduced. The technology used for fabricating 3D ICs will critically impact the BIL and therefore, should be a part of the floorplan optimization problem. Our results indicate that BIL in 3D nets is more sensitive to the variation of TSV resistance with technology, than the number of TSVs assigned to the net. Nanoscale TSVs exhibit remarkable potential in improving the number of buffers per net at any given technology node. Furthermore, at sub-45 nm nodes, nano-scale TSVs can significantly curb the exponential rise in buffer count, thereby improving the silicon area, interconnect delay and power in future 3D IC designs.

5. IMPACT OF TSVs ON PERFORMANCE IN NANO-SCALE 3D CMOS IC TECHNOLOGIES

The projected benefits of 3D ICs such as wirelength reduction and performance improvement, are not just contingent upon the size and number of TSVs, but also influenced by the process technology used to fabricate the 3D ICs [71]. For e.g. a 3μm TSV used in a 3D IC built in 45 nm, will have greater TSV area overhead than inserting the same size of TSV in a 90 nm 3D design. In the ideal scaling scheme, width and height of the global interconnects are scaled keeping the aspect ratio, H/W, constant. Hence, with aggressive technology scaling, the impact on wire resistance is non-linear. Each new technology represents a 1.4x reduction in feature size and a corresponding 1.4-1.5x drop in minimum metal pitch [5]. The impact of TSV size and TSV fabrication techniques on the performance of 3D ICs has been the focus of most works in recent years[39]-[47]. However, the question arises - what is the impact of the continuous advances in nano-CMOS technologies on early estimation of performance and power of future 3D designs ? We seek to address this issue in this chapter.

The usage of nanoscale TSVs is also inevitable in the future nanotechnologies. Due to aggressive scaling and shrinking of wire dimensions, the delay of global wires again poses as the performance limiter in future 3D designs. Additionally, increased capacitive coupling between TSVs and surrounding wires at will further degrade timing of signal transmission [56]. According to projections by the ITRS [4], device downscaling beyond 22 nm node, will also be accompanied by downscaling of TSVs to sub-micron levels, in an effort to curb the adverse impact of TSVs. Hence, there is an imminent need for thorough research into the implications of future nanotechnologies on the quality of 3D designs built using nanoscale TSVs. This needs to be evaluated preferably during early stages of design such as floorplanning. In essence, the success and viability of future 3D IC designs, undoubtedly depends on how early and accurately we predict the influence of future nano-CMOS technologies and nanoscale TSVs on 3D on interconnect performance and power in 3D IC.

5.1. Previous Works

Xu *et al* analyzed the scaling issues with TSVs filled with Cu, W and CNTs [91]. TSV performance with various geometries and different driver sizes at 32nm and 22nm technology nodes was compared. Although a thorough perspective of technology scaling on TSV parasitics was presented, optimization of buffered interconnect delay in 3D IC as a whole was ignored. In [71] the authors study the impact of nano-scale TSVs on area, wirelength, delay, and power on quality of 3DICs. However, timing was optimized on a generated 3D layout with fixed TSV positions, limiting their solution. The complex issue of buffer insertion in 3D nets was not considered. Kim *et al.* [78] first considered the significant TSV RC parasitic contribution towards delay of buffered 3D interconnects. However, they used a fixed interval between consecutive buffers irrespective of number of TSVs in a net, which is unrealistic when technology parameters change. TSV positions were not considered during buffer estimation, resulting in unnecessary insertion of buffers around TSVs. In [76] Lee *et al*. considered TSV RC parasitic delay and TSV positions to find optimum buffer locations in a 3D net. However, they used a fixed layout with fixed TSV positions. The significant coupling between adjacent TSVs on TSV capacitance was also ignored.

None of the previous works consider the combined impact of technology scaling with nano-scale TSVs on early optimization of interconnect delay at 3D floorplanning level. The impact of emerging new TSV fill materials on buffered 3D interconnects has not been considered earlier. Additionally, none of the works consider optimizing TSV distribution on layout which critically impacts number of buffers, delay and power.

In this chapter we make the following contributions:

- We investigate the impact of three diverse nano-CMOS technology nodes, yet to be realized for future 3D ICs - 45nm, 32nm and 22nm, on early estimation of delay, buffers and dynamic power. For each technology node, we consider two different TSV diameters - 3.0µm and 0.5µm, to gain more insight into the impact of nanoscale TSVs on performance and power of future 3D ICs.
- Additionally, this work explores five different alternative TSV fill materials. Their potential effect on the critical distance between consecutive buffers (buffer insertion length) is investigated.

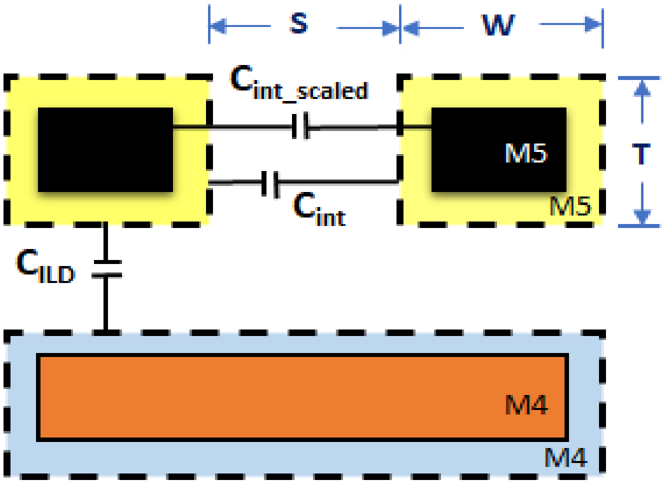


Figure 35. Impact of interconnect scaling on wire capacitance components

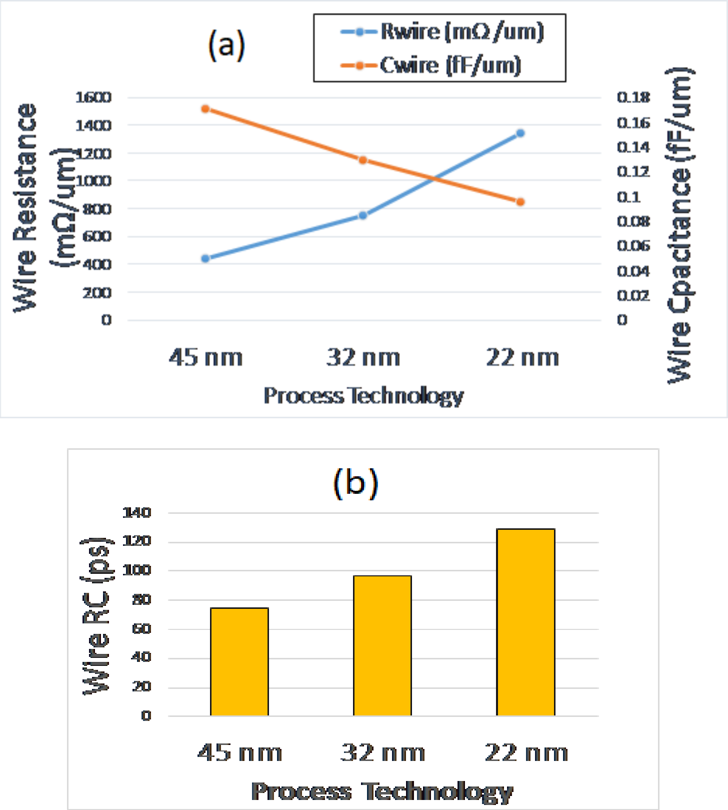


Figure 36. (a) Variation of unit length wire resistance and capacitance with technology scaling (b) Variation of wire RC delay with technology scaling. Wire specifications followed for semi-global (M4-M6) metal layers at each technology.

5.2. Impact of Interconnect Scaling on 3D Design Quality

5.2.1. Wire Resistance and Capacitance

The typical backend structure of integrated circuits which is used to connect the active devices and regions is shown in Fig. 37. The backend structure consists of contacts, vias and interconnects separated by dielectric layers. Interconnects can be global, semi-global or local. Local interconnects made of polySi, TiN or W are used to connect at the device level. As they run shorter distances, higher resistivity materials can be used if they present other desirable properties. They may also serve as the gate electrode material. The semi-global interconnects made of less resistive Al or Cu, serve to connect the devices within a block. Global interconnects traverse long distances connecting the blocks, including power, ground and clock signals. As the dimensions of devices continue to shrink, so are the height and width of the interconnects. In the nanometer era, the relative importance of interconnect structures has increased. The primary reason is the increased circuit area which increases the disparity between the dimensions of the devices and the semiglobal/global interconnects. This phenomenon is demonstrated by a simple schematic of an interconnect structure as shown in Fig. 37. The two metal lines of dimensions of length 'L', width 'W' and thickness 'T' are separated by spacing 'S' which is filled with a dielectric SiO_2 layer. The two

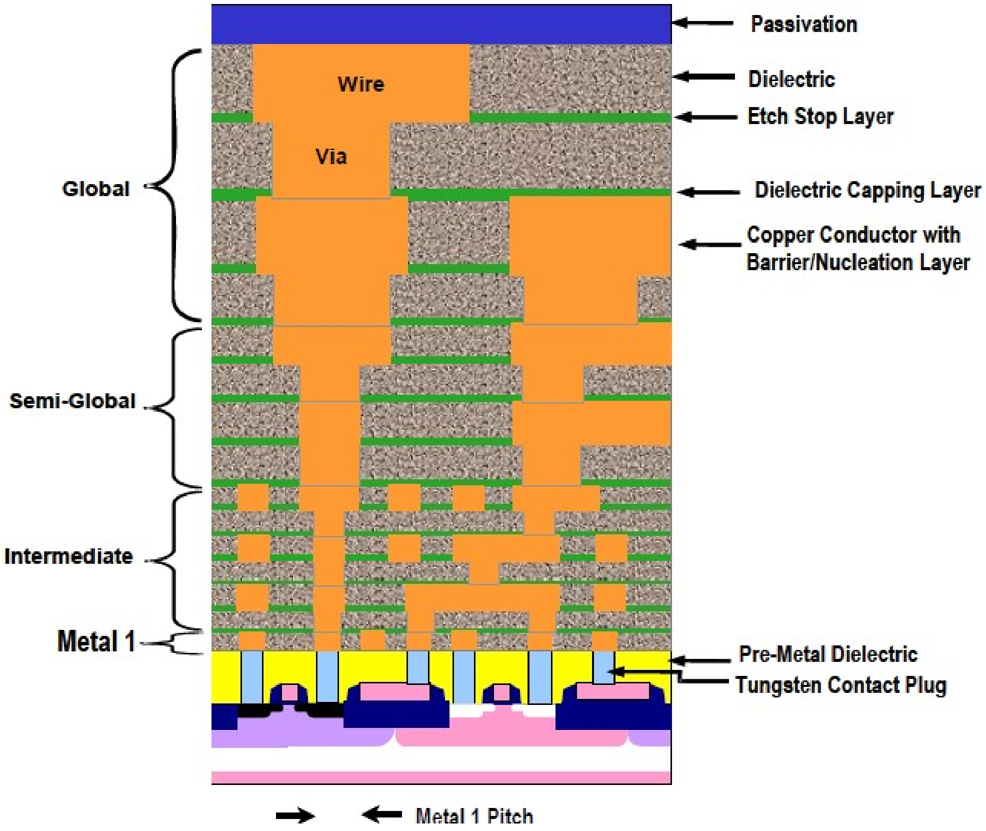


Figure 37. Diagram of backend (BEOL) structure of VLSI circuit. (Source: ITRS [4])

interconnect lines also sit on the top of the dielectric layer of SiO_2. In this case, the line resistance of each interconnect with resistivity ρ is given by Eq 1:

$$R_{wire} = \rho * \frac{L}{W*T} \tag{1}$$

Assuming 'Kox' as the thickness of dielectric between interconnects and ε_0 as the permittivity of the free space, the line-to-line capacitance C_{ILD} between two consecutive layers of interconnects is separated by distance 'X_{ox}'given as:

$$C_{ILD} = K_{ox}\varepsilon_0 * \frac{W*L}{X_{ox}} \tag{2}$$

The capacitance between interconnects on the same layer 'C_{int}' is given by.(3) :

$$C_{INT} = K_{ox}\varepsilon_0 * \frac{T*L}{S} \tag{3}$$

The total time delay of the interconnect structure 'τ_L' is approximately equal to 0.89 RC. Hence,

$$\tau_L = 0.89\, K_{ox}\varepsilon_0\rho\frac{L^2}{W*T}\left(\frac{W}{X_{ox}} + \frac{T}{S}\right) \tag{4}$$

The dimensions of 'S' and 'T' are closest to smallest feature size 'λ' dictated by the lithography and etching capabilities of a technology node. As technology progresses, both X_{ox} and T shrink with λ', keeping the aspect ratio T/W constant. To see what happens to time delay of global and semi-global interconnects, we consider that X_{ox} and T are equal to λ'. Then substituting in Eq.4, It can be observed that for global interconnects, there will be rapid increase of interconnect RC due to shrinking of λ' . This leads to not only performance loss from interconnect delay increase, but circuit power and area degradation as well.

5.2.2 Impact on Wire RC Delay

In the scope of this work, we consider three different technologies – 45 nm, 32 nm and 22 nm to compare the variation in wire RC parameters at the semi-global level. Table 16 shows the Intel's specifications for semi-global metal layers dimensions for the three nodes, which are used to compute the unit wire resistance and unit wire capacitance for each technology. In our approach, the computation of wire capacitance is based on analytical model in [58]. The width and spacing of wire are taken as half metal pitch (MP) specified for each technology.

Fig. 35 illustrates the components of line-to-line wire capacitance before and after scaling of wire dimensions. Fig. 36(a) shows that reduction in wire cross-section significantly increases the wire resistance (3x) from 45nm to 22 nm. On the other hand, the unit wire capacitance decreases by (1.8x) due to decrease in both sidewall capacitance with adjacent wires and the inter-layer capacitances with top and bottom metal. Fig. 36(b) shows the wire RC delay rises dramatically rises by 72% from 45 nm to 22 nm, due to the dominating impact of wire resistance.

5.3. Impact of Technology Scaling on TSV Capacitance

We consider three different TSV sizes – 3um, 1.5um and 0.5um with aspect ratio (height/diameter) of 10, as specified by the ITRS [4]. Two different cluster capacities are considered – (a) small-sized cluster with four TSVs and (b) large cluster with 30 TSVs. It is to be noted that the cluster aspect ratio (array size) impacts the TSV-to-TSV coupling (C_{TT}), whereas change in wire

Table 16. Metal layer (M4-M6) specifications and technology parameters for three different process nodes followed in our experiments.

Parameter	45 nm	32 nm	22 nm
Metal Pitch (nm)	280	225	160
Width (nm)	140	112.5	80
Thickness (nm)	252	204	160
Spacing (nm)	140	112.5	80
Unit wire resistance (mΩ/μm)	487.5	749.5	1343.8
Unit wire capacitance (fF/ μm)	0.171	0.129	0.096
Supply Voltage (V)	1	0.9	0.8
Operating Frequency (GHz)	2	2.3	2.6

Table 17. TSV Specifications for each size considered in our experiments

Diameter (D)	**0. 5μ**	**1.5μ**	**3.0μ**
Height (H)	5μ	15μ	30μ
Aspect Ratio (H/D)	10	10	10
KOZ (μm)	1.0	2.5	4.8
Pitch (μm)	1.0	3.0	6.0
TSV Resistance (mΩ)	438	146	73

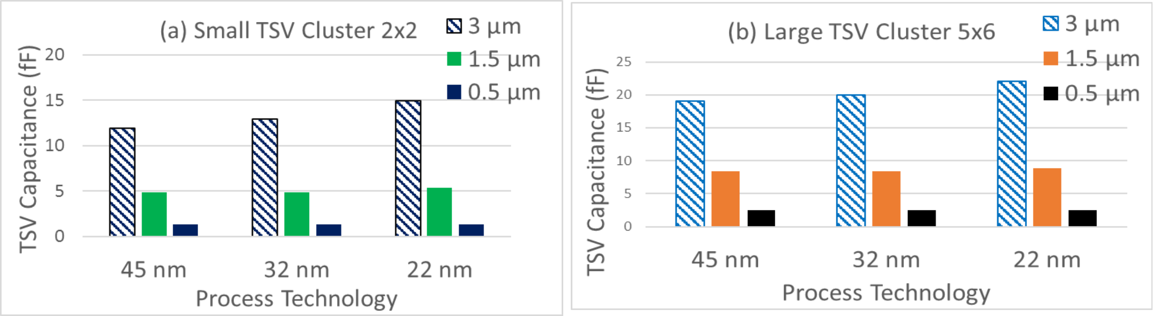


Figure 38. Impact of process scaling on TSV capacitance for three TSV sizes: (a) Small Cluster (b) Large-sized Cluster

dimensions due to technology will influence the TSV-to-wire coupling (C_{TW}). The scaling of TSV diameter will impact both C_{TT} and C_{TW} [Refer Section 3.2.2]. Figure 38(a) and (b) show the impact of downscaling from 45 nm to 22 nm on TSV capacitance (C_{TSV}) for each

TSV diameter. With technology scaling from 45 nm to 22nm, C_{TSV} increases for all TSV diameters, due to the 39% rise in TSV-to-wire coupling (C_{TW}) as the spacing between wire-to-TSV decreases with technology. The 3μm TSV shows larger percentage increase in capacitance (25%) as compared to 1.5um (10%) and 0.5um (1.4%). This is because of larger surface dimensions (top and sidewall of TSV) coupling with wires for larger-sized TSVs. This further demonstrates that in advanced technology nodes, the use of nanoscale TSVs will be critical in minimizing the TSV capacitance contribution. With TSV downscaling, C_{TSV} significantly drops by average 81% for both cluster sizes. The selection of TSV cluster size is also important. Comparing Fig. 38(a) and (b), on average, the TSV capacitance contribution from a small size cluster is 36% lesser than a TSV from a large cluster, for all considered TSV sizes.

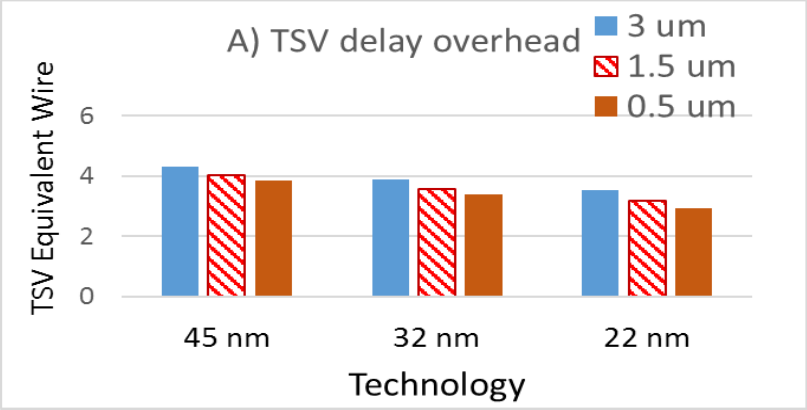


Figure 39. Impact of technology scaling on TSV delay contribution to wire delay

In Fig.39, we compare the relative contribution of TSV to wire delay, represented by the parameter TSV equivalent wirelength discussed in [47]. In general, at all nodes, with decreasing TSV size, the TSV delay contribution to wire goes on decreasing due significant drop in TSV capacitance (81%), which dominates the decrease in wire capacitance (44%). For a given TSV size, with technology scaling, the TSV RC remains constant, but wire RC rises. Hence, TSV contribution to wire delay decreases.

5.4. Experimental Results

5.4.1. Impact of TSV Downscaling and Technology Scaling

Tables 18,19 and 20 show the impact of scaling technology and scaling of TSV dimensions on the evaluated interconnect delay and power. With scaling of TSV dimensions from 3µ to 0.5 µ, a significant reduction in TSV area and capacitance reduces total delay by 11% at 45 nm, 14% at 32 nm and by 14% at 22nm. With technology scaling, the total delay increases by 52% for the largest TSV size 3µm, while it rises by 48% for 0.5µm, primarily due to increase in wire resistance and rise in coupling between wire and TSVs. With the scaling of TSV diameter from 3.0 μm to 0.5 µm, the rise in delay with technology also lessens, as the TSV delay contribution relative to wire delay steadily decreases.

For all TSV sizes, the estimated total interconnect power reduces with technology scaling due to reduced wire capacitance and supply voltage. For 3µm TSV, the power reduces on average by 37% scaling from 45 nm to 22 nm. With TSV scaling to 0.5µm, the power reduction rises to 53%. With advancing technology, interconnect power is dominated by TSV power, as the wire capacitance progressively decreases. Therefore, with downscaling of technology in nanometer range, the minimization of TSVs in individual wires during nets-to-TSVs assignment will be even more important than before. At the same time, selection of appropriate size of TSV cluster for assignment of nets will also be critical for minimizing the TSV power

Table 18. Comparison of interconnect delay and power for three nodes. TSV D = 3μ, H = 30 μ, Pitch = 6 μ and KOZ = 4.6 μ.

Circuit	45 nm			32 nm			22 nm		
	Area (mm^2)	Delay (ps)	Power (mW)	Area (mm^2)	Delay (ps)	Power (mW)	Area (mm^2)	Delay (ps)	Power (mW)
N100	0.226	846	5.1	0.221	1036	4.0	0.225	1488	3.2
N200	0.281	1857	11.5	0.285	2077	9.2	0.277	2477	7.5
N300	0.461	2491	15.8	0.477	2731	12.6	0.478	3668	10.1
		1.0	**1.0**		**1.15**	**0.78**		**1.52**	**0.63**

Table 19. Comparison of interconnect delay and power for three nodes. TSV D = 1.5μ, H = 15 μ, Pitch = 3 μ and KOZ = 2.5 μ

Circuit	45 nm			32 nm			22 nm		
	Area (mm^2)	Delay (ps)	Power (mW)	Area (mm^2)	Delay (ps)	Power (mW)	Area (mm^2)	Delay (ps)	Power (mW)
N100	0.223	821	4.0	0.220	994	2.9	0.221	1325	2.1
N200	0.278	1801	8.6	0.277	1993	6.3	0.273	2204	4.7
N300	0.459	2416	12.1	0.463	2622	8.9	0.474	3261	6.4
		1.0	**1.0**		**1.13**	**0.73**		**1.39**	**0.53**

Table 20. Comparison of interconnect delay and power for three nodes. TSV D = 0.5μ, H = 5 μ, Pitch = 1μ and KOZ = 1.0 μ

Circuit	45 nm			32 nm			22 nm		
	Area (mm^2)	Delay (ps)	Power (mW)	Area (mm^2)	Delay (ps)	Power (mW)	Area (mm^2)	Delay (ps)	Power (mW)
N100	0.22	816	3.4	0.219	875	2.4	0.22	1219	1.6
N200	0.276	1638	7.1	0.273	1875	4.9	0.274	1998	3.3
N300	0.452	2297	10.1	0.451	2359	7.1	0.455	2997	4.7
		1.0	**1.0**		**1.09**	**0.69**		**1.33**	**0.47**

5.4.2. Impact of Technology on Buffer count, Total delay and Power

Fig 40 shows the impact of technology scaling from 45 nm to 22 nm on buffer count for all three benchmarks. We compare the results between two different TSV sizes -3 μm and

0.5µm. As technology advances to 22 nm, the total buffer count increases on average by 26% for 3µm and by 19% for 0.5µm. The increase in buffer count is attributed to dramatic (78%) increase in wire RC delay with technology scaling and rise in delay of 3D segments due to (38%) increase in TSV capacitance [73].

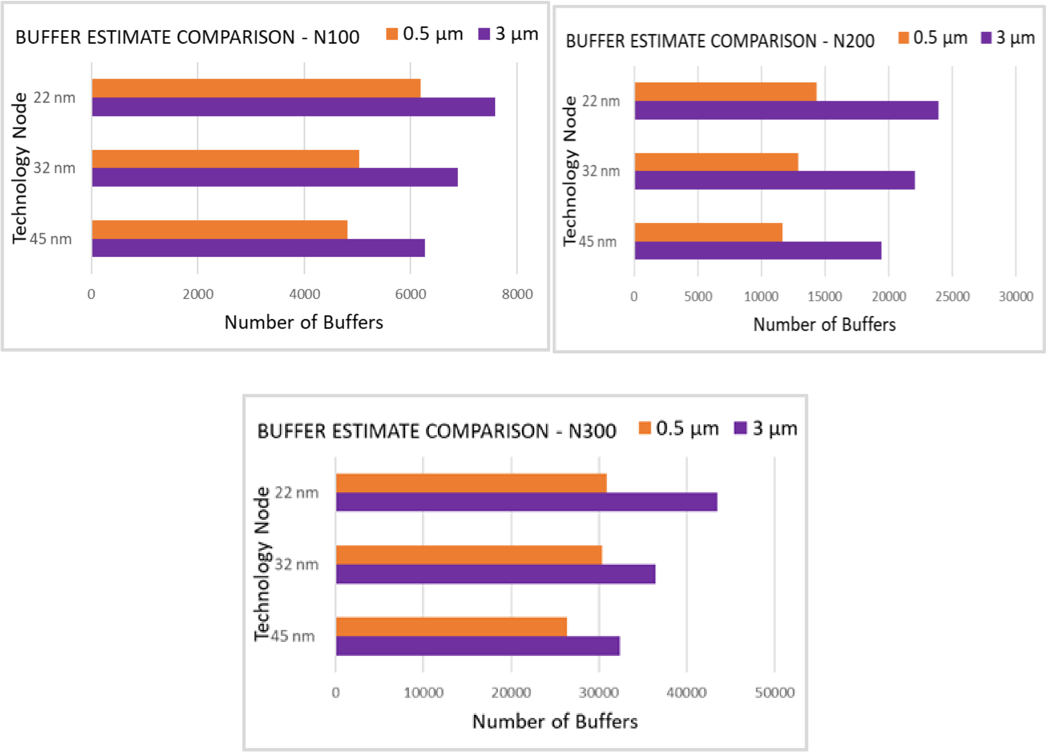


Figure 40. Variation of total buffer count with technology scaling and TSV downsizing for three GSRC benchmarks

The interesting impact of nanoscale TSV is also observed here. With 0.5µm TSV, the rate of increase of buffer count with technology scaling is smaller compared to 3µm TSV. This is due to significant reduction in TSV capacitance (85%) with TSV downsizing, which decreases the overall delay of 3D segments and hence decreases number of buffers per net. With TSV downscaling to 0.5µm, buffer count reduces by 27 % for 45 nm, 29% for 32 nm and 31% for 22 nm. This is due to increase in TSV resistance with smaller TSV diameter, which lengthens the required BIL, reducing the average buffer count per net.

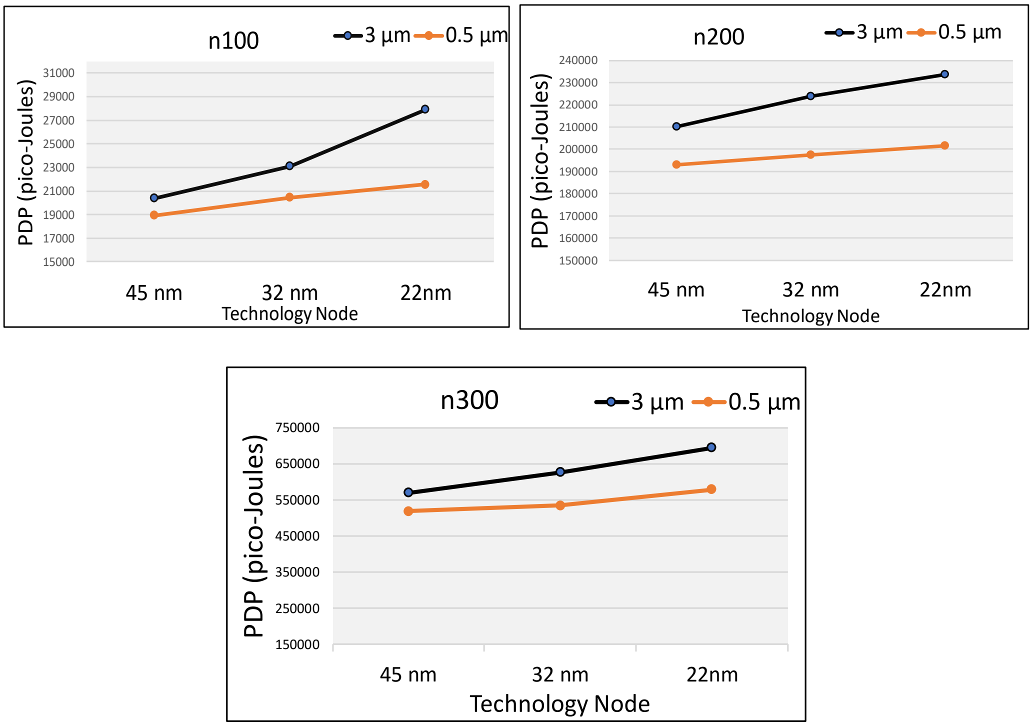


Figure 41. Power-delay product (PDP) variation with technology scaling for two TSV sizes and three GSRC benchmark circuits

5.4.3. Comparison of power-delay product

We use the metric power-delay product (PDP) for each benchmark to capture the combined trend of interconnect delay and power with technology scaling. Fig 41 shows the variation in PDP for two TSV diameters, for three GSRC benchmarks. Migrating from 45 nm to 22 nm, PDP increases with technology scaling by average 24% for 3 µm and average of 11% using 0.5 µm TSV. With nano-scale TSV, the PDP is average 11% lesser than 3 µm TSV, for all benchmarks. The rise in PDP with technology scaling for both TSV sizes is mainly due to significant increase in total delay (average 36% for 3µm and 16% for 0.5 µm TSV for all benchmarks), due to the cumulative impact of increase in buffer count, wire delay and TSV capacitance. The reduction in rate of PDP increase in case of 0.5 µm TSV, is due to its significantly smaller rate of increase in TSV capacitance with technology, as compared to 3µm TSV [73]. With scaling from 45 nm to 22 nm, total interconnect dynamic power decreases by average 8% for 3µm TSV and by 13% in 0.5 µm TSV for all benchmarks, primarily due to reduced operating voltage and significant decrease in wire capacitance. TSV downsizing to 0.5 µm also leads to reduction in both buffer count per net and TSV capacitance, significantly reducing interconnect power by average 11% for all benchmarks. This further shows the positive impact on chip performance in lower nodes, when using nanoscale TSVs.

5.4.4. Influence of TSV Material Resistivity on Buffers

We examine the impact of five different TSV fill materials – copper (Cu), tungsten (W), copper-carbon nanotube (Cu-CNT) composite, nickel (Ni), and single-walled carbon nanotube bundle (SWCNT) on the estimated buffers in 3D ICs. Table 21 compares the above materials on the basis of their important electrical and thermal properties as reported in previous works [89]-[99]. Traditionally Cu has been used for TSV filling due to its high conductivity and economic feasibility [91]. Electroless plating of Cu has also been recently demonstrated for 3D applications [93]. However, Cu is susceptible to signal distortion at higher frequencies (>10 GHz) due to skin effect and issues of thermal instability. Large

Table 21: Electrical and thermal properties of TSV conducting materials

TSV fill Material	Resistivity (Ω.m)	Thermal Conductivity (W/m.K)	Coefficient Thermal Expansion at 20ºC (ppm/ºC)
Copper (Cu)	1.68 x 10^{-8}	400	16.5
Tungsten (W)	5.6 x 10^{-8}	175	4.6
Nickel (Ni)	7 x 10^{-8}	91	13.3
Cu-CNT composite	3.36 x 10^{-8} [95]	800 [90]	5
SWCNT bundle [94]	6.17 x 10^{-8}	1767	-0.1

mismatch of coefficient of thermal expansion with Si (2.6 ppm/ºC) induces high tensile stress [81]. Also, Cu has severe electromigration issues under high current density state and is difficult to fill high aspect ratio vias [99]. Tungsten (W) TSVs offer an interesting compromise to Cu-TSV mainly due to well established compatibility with the CMOS transistors and very good step coverage of high aspect ratio via with commonly used WF6 based CVD chemistry [94]. With almost no CTE mismatch with Si, W-TSV minimizes the requirement of keep-out zone [81]. Its ability to fill very high aspect ratio vias (~50) is also shown [94]. However, the high stress of the film drastically limits the allowed thickness deposition. Ni is also a favorable candidate for TSVs due to its lower CTE and higher electroplating deposition rate than Cu. According to [95], high aspect ratio (~7.5), void and defect free vias using Ni electroplating have been recently demonstrated.

Carbon nanotubes (CNTs) are recently being explored as a promising alternative for TSV material and are an impressive possibility at relatively lower frequencies (~10GHz) [99]. SWCNT bundles offer excellent performance advantages due to high current carrying capacity, good thermal stability and negligible electromigration enabling a high performance to cost ratio for 3D heterogeneous integration, even at higher frequencies [91]. However, the SWCNT fabrication still suffers from high CNT growth temperatures (for CMOS compatible processing), achieving a good packing density using the available CNT synthesis processes. The smallest packing density predicted for CNT bundles in interconnect applications is ~1014 tubes/cm^2 assuming SWCNT diameter of 0.8nm [96]. In real applications, the electrical resistivity of CNT TSVs is still several orders of magnitude (~48) higher than that of copper TSVs due to their porous structure [97]. Recently, efforts have been made to blend the benefits of pure Cu and pure CNT vias. This has resulted in emergence of Cu-CNT composite TSVs, in which the nanotubes are vertically aligned and directly grown bottom-up in the etched via and filled with Cu using a low current density electroplating process [98]. They are fabricated with almost similar conductivity as Cu, yet 100x higher current carrying capacity than Cu. As Cu current density reaches the breakdown limit with scaling, Cu-CNT composite will be of great demand as TSV filler, being the only material with high conductivity and high ampacity [91].

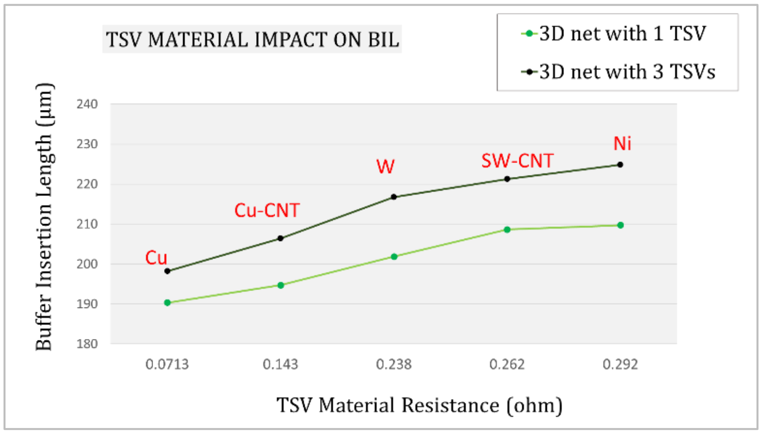


Figure 42. Variation of BIL with TSV material resistance. The impact of TSV material resistivity becomes dominant with increase in number of TSVs per net.

For our analysis, we consider TSV diameter of 3μm and height of 30 μm. We consider 45 nm specifications for unit wire resistance and capacitance values specified in Table 16. For each fill material, we consider the resistivity at room temperature. TSV resistance with each filler material is determined using analytical model in [6]. Buffer insertion length is estimated using (3) for each potential TSV material. Fig. 42 shows the variation of BIL per unit ohm increment in TSV resistance. The slope (Δy/Δx) of the plot shows that BIL increases by 78 μm per unit ohm increase in TSV resistance for 1-TSV net, and by 120 μm for 3-TSV nets. Compared to the base case of Cu-TSV, BIL increases by 17% for SWCNT bundle TSV, 13% with Ni-TSV, 7% with W-TSV and only ~2% increase with Cu-CNT composite. The almost comparable BIL estimate and conductivity for Cu and Cu-CNT composite, indicates the promising future applications of Cu-CNT, as a direct alternative to Cu TSV, especially in the high frequency RF range where Cu suffers from numerous reliability challenges. Compared to Cu, the higher BIL obtained with SWCNT-TSVs would lead to lesser number of buffers per net. With rising wire delay in advanced nodes, the intrinsic delay of buffers decreases however number of buffers increases. In this scenario, use of SWCNT-TSVs will be beneficial in countering the dominant buffer delay.

5.5. Chapter Conclusions

We investigated the impact of three diverse nano-technologies and nano-scale TSVs on early performance and power estimation in nano-CMOS 3DICs. Using nano-scale TSVs offers significant reductions in buffer count, delay and power, primarily due to reduced TSV capacitance, showing promising applications in improving design quality in future nano-technologies. Most notably, compared to micron-sized TSVs, using nano-scale TSVs successfully curb the rate of increase of buffer count with advancing technology, resulting in significantly improved power-delay product at 22nm. This shows the excellent potential of nano-scale TSVs in building of high-performance and energy efficient 3D circuits at sub-45 nm technology nodes.

In addition to this declining trend in TSV sizes, exploration of new and alternative TSV fill materials to Cu, will play a key role in minimizing the dominant setbacks due to interconnect delay. The choice of TSV material directly impacts the buffer insertion length and buffer count per net, both key determinants of interconnect delay and power. As TSV resistance escalates with TSV scaling, CNTs can be a potential alternative if the CNT bundle density is improved and growth temperatures are minimized to integrate into the conventional CMOS process. The negative coefficient of thermal expansion of CNTs can also aid in minimizing the keep-out-zone of TSVs still withstanding the TSV-induced stress and saving significant silicon area. SW-CNT TSVs with high resistance will result in a lower buffer count compared to any other TSV material. However, to reap the benefits of SW-CNTs, their domination of buffer and wire delay should decline. High-density CNTs should therefore be maintained within the bundle to provide a greater electrical and thermal conductivity than Cu or W TSVs.

As number of buffers in future nano-scale 3D ICs is likely to rise significantly with technology, early and accurate estimation of buffers will be highly consequential in systematically improving the timing and power estimates in the later stages. An important takeaway is that the delay a buffered 3D wire has to be accurately characterized, considering both TSV RC parasitics and TSV position along the net. This also helps eliminates the need for insertion of extra buffers around TSVs, resulting in more effective optimization of delay and power.

6. TSV REDUNDANCY-AWARE 3D FLOORPLANNING

One of the major roadblocks towards successful adoption of 3D ICs is the uncertainty of their stability and reliability, which can potentially incur a significant loss of yield [73]. The increased power density and the complexity of TSV manufacturing process are two major contributing factors which significantly increase the risk of failure of 3D ICs, therefore, degrading their reliability.[80]. Recent studies conducted towards improving the reliability of 3D ICs have focused on thermal-aware 3D design flows [81][82][84], which aim at minimizing the power density in 3D systems. However, the thermal-aware design methods only pertain to failure of 3D ICs at the time of execution. It is equally vital to pro-actively address the possibility of 3D chip failure due to failure of TSVs occurring in the bonding or manufacturing process [85]. As only one failed TSV can be enough to fail the entire chip even with all known-good dies, it is now inevitable to incorporate an efficient repair mechanism to recover failed TSVs, preferably in the early design stage. An efficient solution to recover failed TSVs and improve the yield of 3D ICs, is the addition of spare or redundant TSVs in 3D IC design [86][87]. The goal of TSV redundancy is to develop fault-tolerant TSV structures such that signal paths from failed functional TSVs (f-TSVs) in a TSV block or structure can be efficiently re-routed to non-defective spare TSVs (s-TSVs) in the same structure, without incurring a major wire or delay overhead [Figure 43].

6.1. Previous Related Works

Nain et al. [101] attempted to improve the yield by providing wireless redundant TSVs,and performed Monte Carlo simulation under different TSV defect rates to estimate the chip yield. However, huge extra costs including transmitter and receiver circuits may be introduced to ensure the functionality of the employed wireless redundant TSVs. Zhao et al. [14] tried to determine the Hsieh and Hwang [84] presented a repair mechanism, which partitions f-TSVs into TSV groups and assigns each TSV group with one s-TSV for repairing the faulty link in that TSV group. Jiang et al. [100] proposed a TSV redundancy architecture using dedicated switches to handle clustered TSV faults. The proposed technique enables faulty TSVs to be repaired by s-TSVs that are distant rather than by the neighboring s-TSVs, thus being suitable for repairing the clustered TSV faults. Loi *et al.* [102] proposed a ring-based redundant TSV architecture, which places s-TSVs at the edges of the f-TSV grid with multiple rings. Simulation results show that the ring-based architecture can efficiently repair clustered faulty TSVs with low area overhead. However, these methods are only suitable for uniform TSV designs, where TSVs are placed in a regular structure on the die. Chen *et al.* [103] studied an optimal assignment of s-TSVs

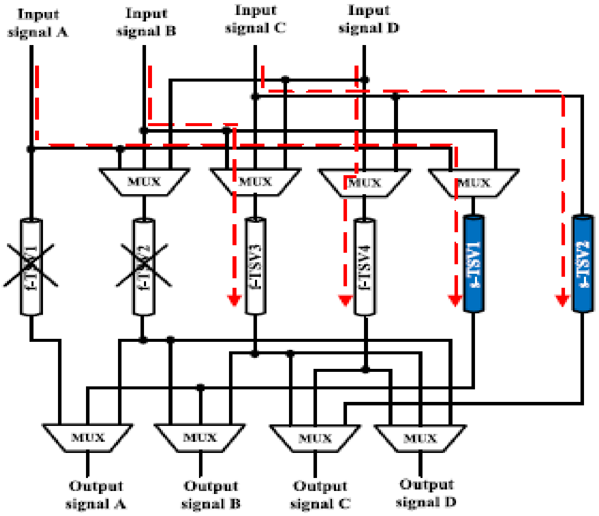


Figure 43. Signal re-routing through s-TSVs 1 and 2 in a fault-tolerant TSV structure when f-TSVs 1 and 2 fail. (Source: [85])

under yield and timing constraints to minimize the total area overhead, where at most one s-TSV can be assigned into a TSV group. Wang *et al*. [104] presented a fault-tolerance technique that can repair faulty TSVs based on a realistic clustered defect model. It showed that the hardware cost is proportional to the number of f-TSV groups, so a greedy algorithm is first used to partition the f-TSVs into several groups with minimizing the hardware cost. Then an ILP-based algorithm is utilized to determine the exact locations of the inserted s-TSVs to minimize the delay overhead.

The common limitation of all of the above works is that they focus on allocation of spare TSVs at a post-floorplanning stage or the placement stage to increase the chip yield, while primarily relying on the availability of whitespace distribution on a fixed floorplan. Hence, the approaches are limited by the quality of floorplan. Some cases may require additional area for allocation of the minimum required spare TSVs. Even if the targeted yield is met, the resulting change in floorplan degrades the overall solution quality. Secondly, the critical impact of additional area and delay overhead of allocation of spare TSVs on chip performance has been completely overlooked.

6.2. Problem Formulation

We define the following important terminologies related to implementation of the TSV redundancy scheme:

- N -> total number of TSVs for benchmark circuit
- N_s -> total number of spare TSVs (to be estimated)
- N_f -> total no. of functional TSVs (after spare TSV allocation)
- total (K_i) - Capacity of a 'i^{th}' cluster 'K_i'.

- s-tsv (K_i)- minimum number of spare TSVs allocated in a cluster ‘K_i’
- f-tsv (K_i) -number of functional TSVs in a cluster after applying redundancy.

Problem Statement - The problem of TSV redundancy is formulated as follows. Given a current floorplan ‘F’ with ‘m’ number of TSV clusters $K_1, K_2 \ldots K_m$ where, $1 \leq i \leq m$ and a uniform TSV failure rate of ‘f ‘. The objective is to estimate the minimum required number of spare TSVs *s-tsv* (K_i) in a cluster ‘K_i’ such that the cluster is fully *repairable,* in case of ‘x’ failed TSVs in the same cluster. Accordingly, we define the condition of repairability as

$$x \leq s\text{-}tsv\,(K_i) \tag{1}$$

i.e. a failed cluster K_i is fully repairable if and only if ‘x’ is less than the allocated minimum spare TSVs.

6.3. Method of Estimation of Minimum Spare TSVs

In order to solve the above problem, we use a probabilistic approach for estimating the minimum required number of spare TSVs in a TSV cluster. We begin with the modeling of TSV redundancy with two important assumptions. Firstly, we assume an *independent TSV defects distribution* model i.e., failure of a TSV in a cluster does not affect the functioning of other TSVs in the same cluster or other clusters containing functional TSVs. The presence of clustered TSV defect distribution is beyond the scope of this work. Secondly, we assume that all TSVs within a cluster have a uniform *failure rate ‘f’.* The number of spare TSVs allocated for a given cluster size is fixed and remains unchanged until the cluster size changes.

Following the above assumptions, the first step is to find the number of possible ways in which a TSV cluster can fail due to failure of ‘*x*’ TSVs. This is represented by the combination of cluster size *total (K_i)* and ‘*x’*, which is expressed by (2):

$$N(x) = C_x^{total(K_i)} \tag{2}$$

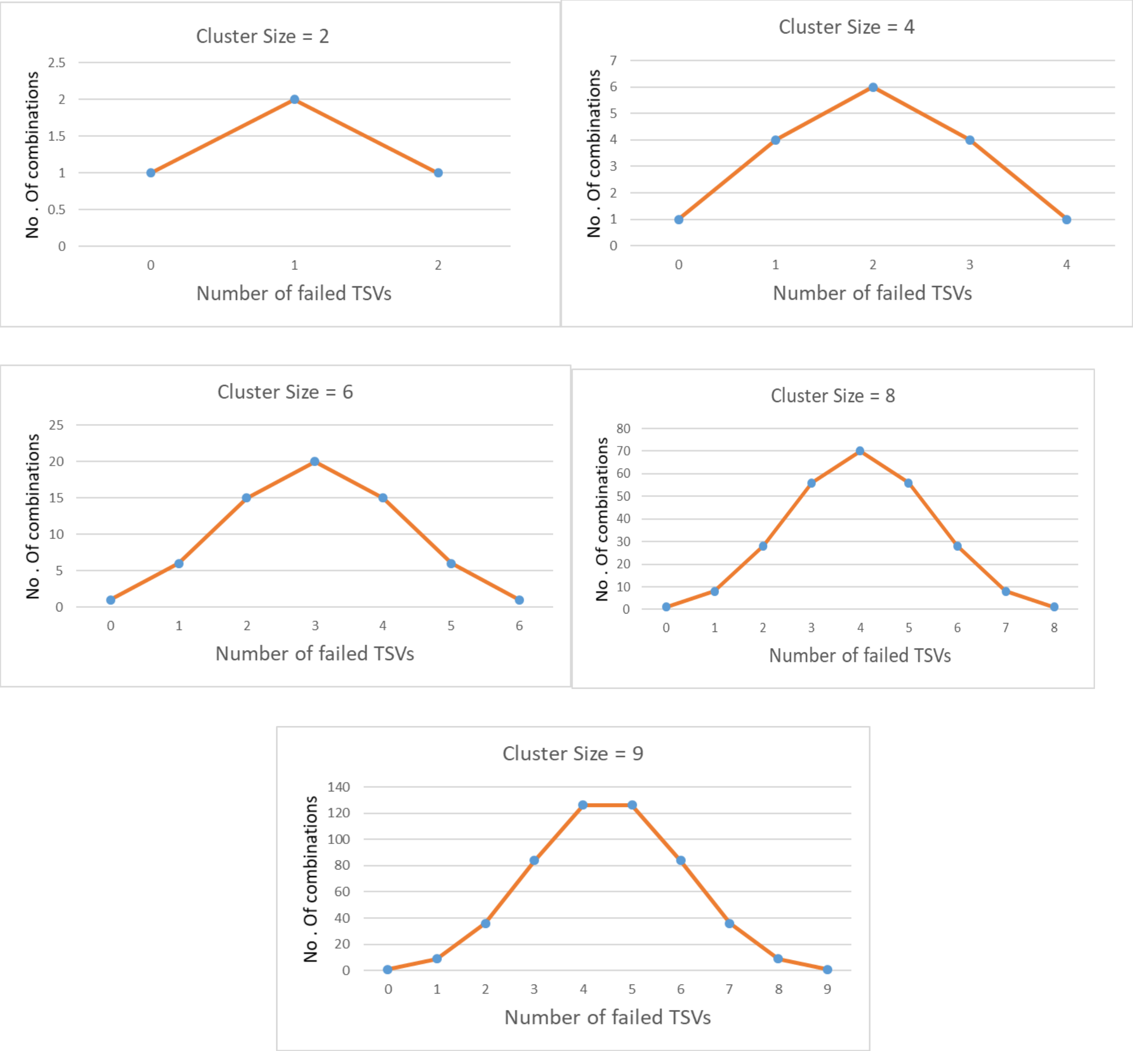


Figure 44. Number of possible situations of ‘x’ failed TSVs in small sized TSV clusters. $0 \leq x \leq total(K_i)$.

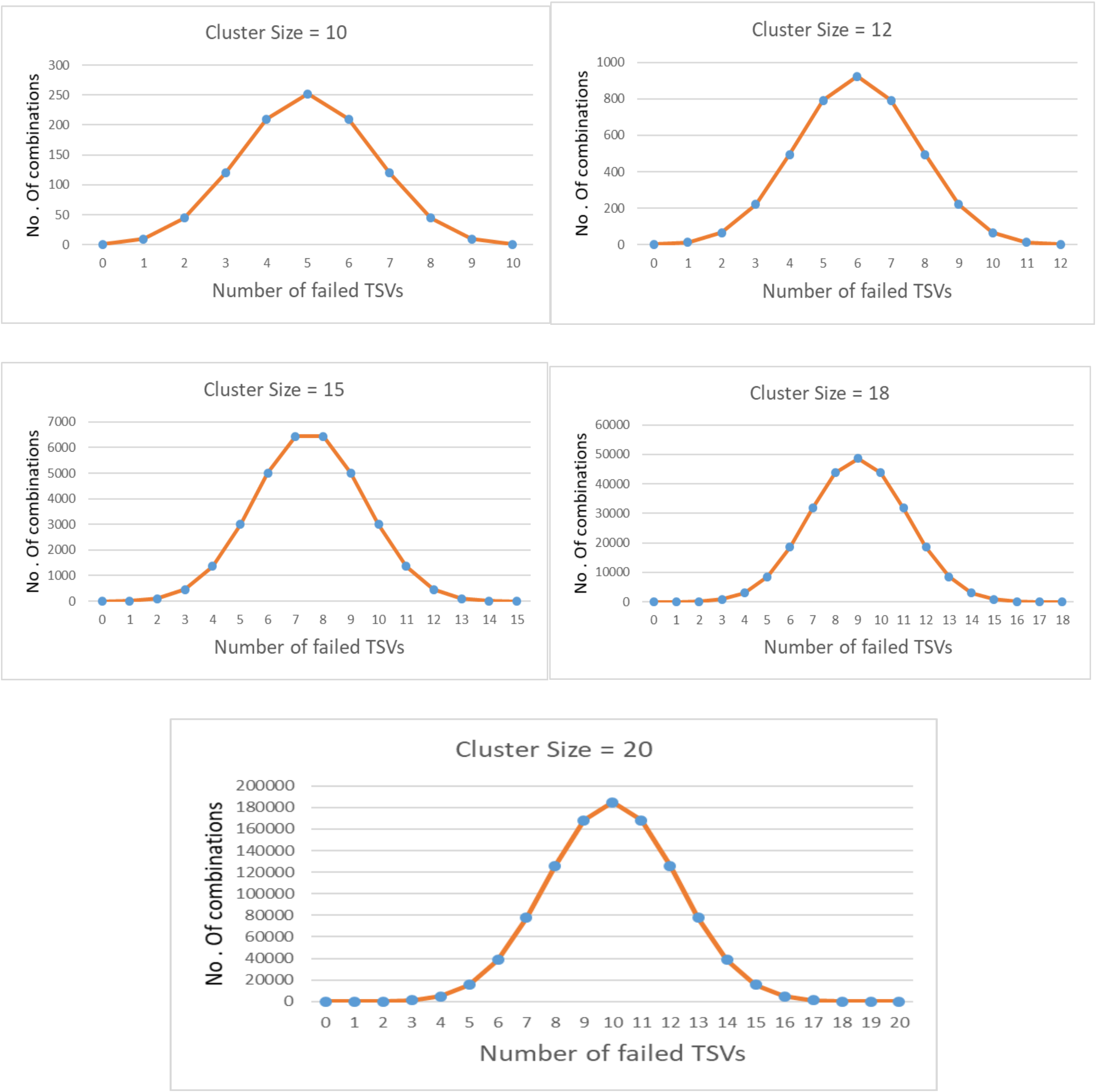


Figure 45. Number of possible situations of 'x' failed TSVs in medium-sized TSV clusters $0 \leq x \leq total(Ki)$.

Figure 44-46 show for all cluster sizes, the number of possible situations of having 'x' failed TSVs when cluster capacity *total* (K_i) is given, such that $1 \leq x \leq total(K_i)$. or small-sized clusters, medium-sized clusters and large-sized clusters respectively. As each TSV within a cluster is independent and has a uniform failure rate '*f*', the number of failed TSVs in a cluster follows binomial distribution. Then, the probability of having '*x*' defective TSVs can be expressed as follows:

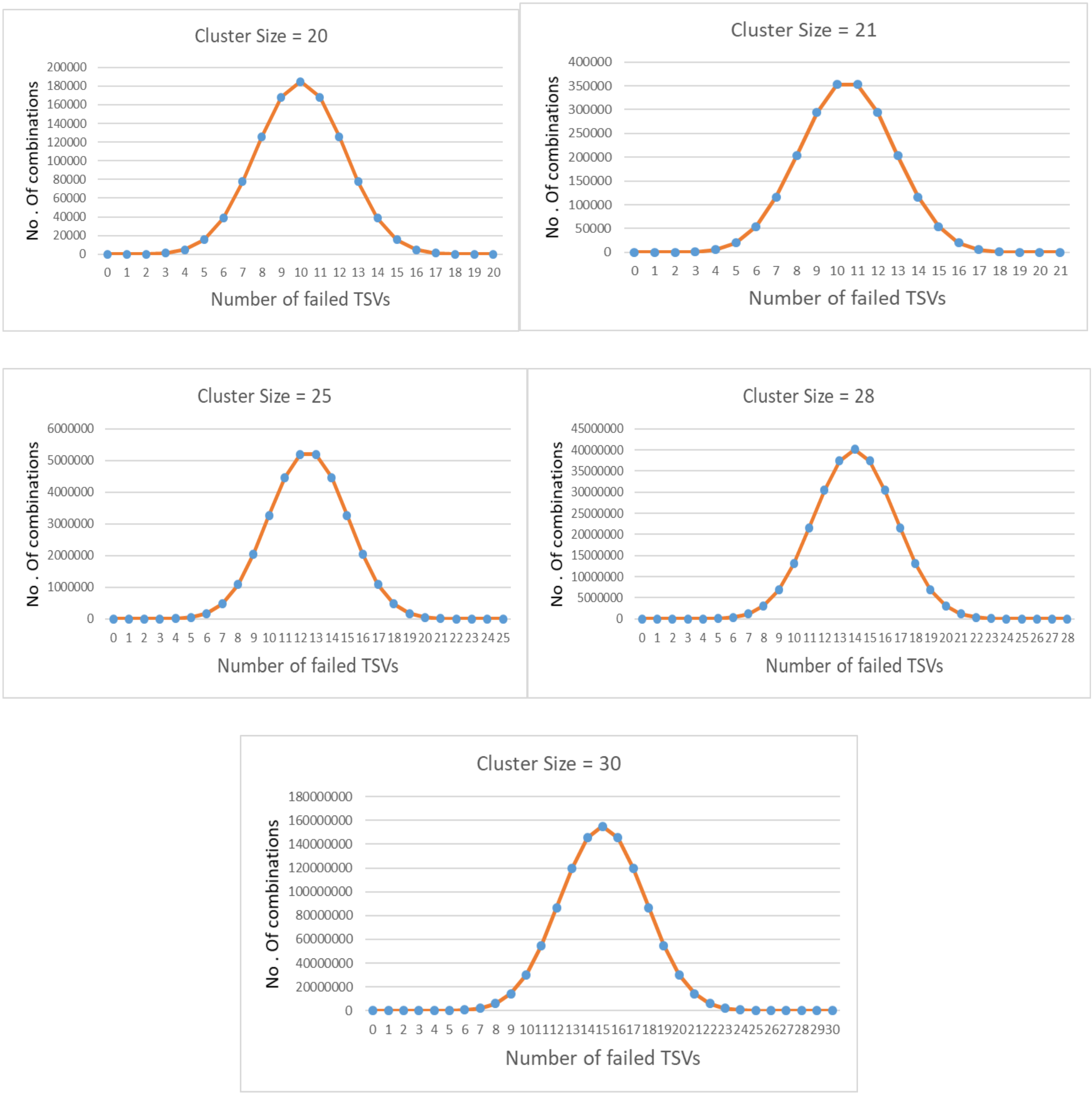


Figure 46. Number of possible situations of ‘x’ failed TSVs in large-sized TSV clusters

$$P_{\text{fail}}(x) = C_x^{total(K_i)} * f^x * \ (1-f)^{[total(K_i)-x]} \tag{3}$$

Here, the product of the second and third term represents - the probability that ‘x’ TSVs in the given cluster are failed and remaining [total(K_i) – x] TSVs are not failed.

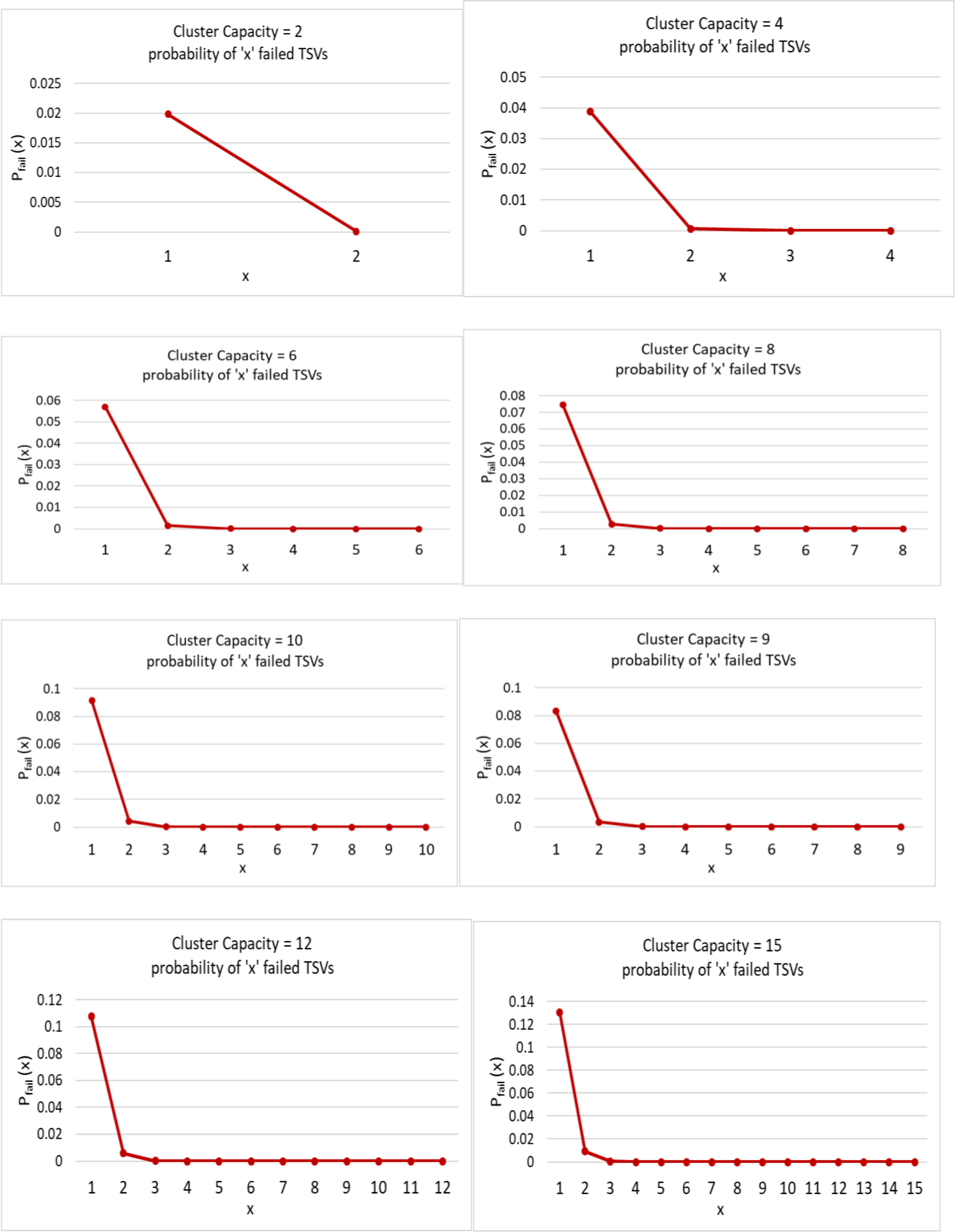


Figure 47(a). Probability of failing of ‘x’ number of TSVs for cluster sizes 2-15, as considered in our experiments.

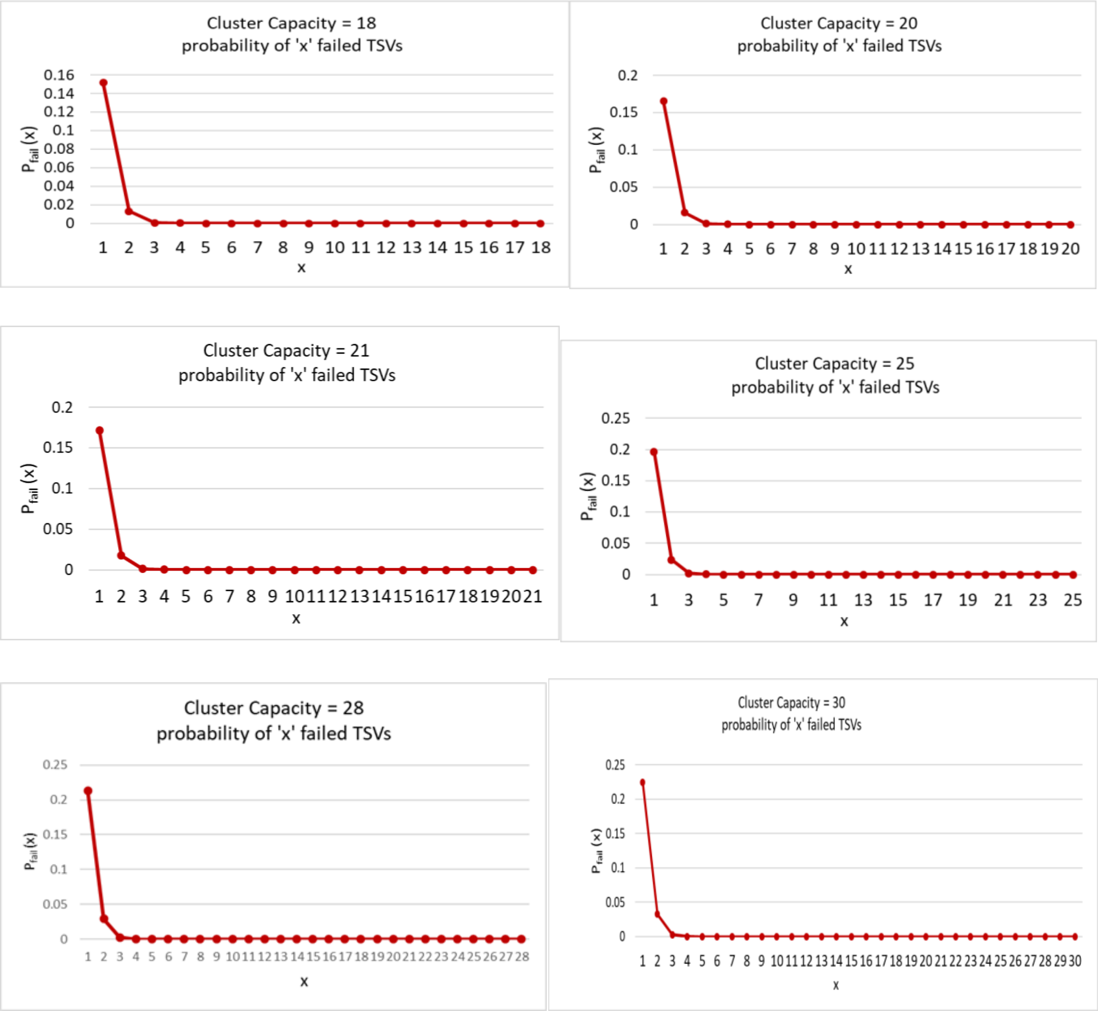


Figure 47(b). Probability of failing of 'x' number of TSVs for cluster sizes 18-30, as considered in our experiments. With increase in cluster size, probability of a single occurrence of failed TSV increases

Fig.47(a)&(b) show the probability of failure of 'x' TSVs in clusters of all sizes considered in our floorplanning. It is observed that the probability of failure of one TSV in a given cluster size is much higher than the probabilities of failure of more than one TSV. Also, the probability of failure of one TSV significantly increases with increase in cluster size. This implies that the minimum required number of spare TSVs is dependent on both the TSV failure rate and the size of TSV cluster. It is worth noting here that, summing up all the probabilities of failure of ‘x’ TSVs in a cluster such that $0 \leq x \leq total(Ki)$, we get the overall yield of a cluster $Y_{cluster}$..Therefore, as long as the cluster yield $Y_{cluster}$. ≅ 1 , the cluster is fully repairable.

The, minimum number of spare TSVs for a given cluster capacity is expressed as:

$$s\_tsv_{min} = P(x) * N_{total} * total(K_i) \tag{4}$$

Where N_{total} is the population of clusters with capacity total Ki, occurring in a given floorplan. We select 15 different cluster capacities to implement the TSV redundancy scheme as shown in Table 22. To construct a standard look-up table we consider the mean of 25 floorplanning runs, to obtain the average value of N_{total} for each cluster capacity as present on the final floorplan. Table 22 reports the estimated minimum spare TSVs for given cluster size as floorplanning input.

Table 22. List of minimum number of required spare TSVs for given cluster capacity

Category	***Small Clusters***					***Medium Clusters***					***Large Clusters***				
Legal Cluster Capacity	1	2	4	6	8	9`	10	12	15	18	20	21	25	28	30
Minimum no. of spare TSVs	0	1	2	3	4	4	5	6	8	8	7	11	10	12	13

6.4. Implementation of TSV Redundancy Scheme

Fig. 48 shows the modified flow of 3D floorplanning, including the TSV redundancy scheme. The algorithm for TSV redundancy is incorporated prior to the nets-to-TSVs assignment, hence, is applied to each floorplan in the current population. This is to ensure that each TSV cluster on the layout is allocated the minimum number of spare TSVs, which are not used during nets-to-TSVs assignment.

The sequence of steps for implementing TSV redundancy are as follows:

- The input to the algorithm consists of the floorplan 'S' consisting of 'n' device layers, a map 'Vn' storing all *valid clusters* on each device layer in except the bottom-most device layer, a 2D array 'Un' which records the number of TSVs assigned for each *valid cluster.*

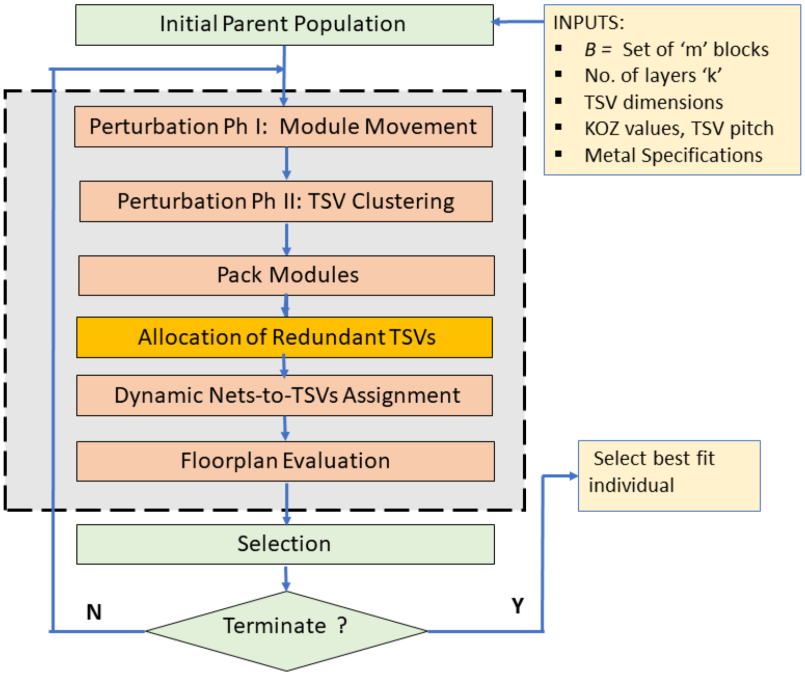

Initial Parent Population
INPUTS:
B = Set of 'm' blocks
No. of layers 'k'
TSV dimensions
KOZ values, TSV pitch
Metal Specifications
Perturbation Ph I: Module Movement
Perturbation Ph II: TSV Clustering
Pack Modules
Allocation of Redundant TSVs
Dynamic Nets-to-TSVs Assignment
Floorplan Evaluation
Select best fit individual
Selection
N
Terminate ?
Y

Figure 48. TSV redundancy-aware 3D floorplanning with dynamic TSV clustering. The allocation of spare TSVs precedes the nets-to-TSVs assignment process in each iteration.

(A TSV cluster is considered valid if it is legal and has at least one TSV available for assignment to a net)

- Each element of the map 'Vn' comprises of a 'key' which stores the index of the valid cluster 'K_i' and an associated 'value' which stores the capacity of the cluster 'total_K_i.

Using the standard look-up table (Table 22) for TSV redundancy, the map '*Vn*' is updated. The current capacity of each valid cluster on a layer is modified to a new capacity such that

$$Total\_(Ki) = functional\_(Ki) + stsv_{min}\,(Ki) \qquad (5)$$

Where, *functional_(Ki)* represents the number of regular or functional TSVs in the cluster 'K_i' and $stsv_{min}$ represents the number of spare TSVs in the cluster. *$stsv_{min}$ (Ki)* denotes the number of spare TSVs allocated to the cluster. A dynamic 2D array *spare_tsv[][]* records the index of the valid cluster and the corresponding number of spare TSVs allocated to it.

Conditions for TSV Redundancy Implementation –

- A TSV cluster is considered *invalid* when all *functional* TSVs in that cluster are assigned to nets. In that case, it is deleted from the map 'Vn'.
- A single TSV if available at the end of nets-to-TSVs assignment, is declared as an independent spare TSV and gets added to the array spare_tsv[][] at the end of nets-to-TSVs assignment.

6.5. Experimental Results

6.5.1 Impact on TSV Usage

As discussed in Section 3.4, the maximum number of TSVs for a benchmark circuit is estimated based on a probabilistic model [47], which considers the number of pins in a net and the number of device layers. However, not all TSV get assigned during the nets-to-TSVs assignment process. These TSVs contribute to the packing area while remaining *unused* on the final floorplan as part of the whitespace. To overcome this limitation, one of the goals of implementing TSV redundancy in the current version of the floorplanner is to increase the usage of TSVs. Under this scheme, the aim is to allocate maximum possible number of unassigned TSVs as *spare TSVs* in individual clusters. This aids in minimizing the number of *unused* TSVs on the floorplan. Table 23 shows the comparison of total number and percentage of TSVs remaining *unused* at the end of floorplanning for two cases : i) floorplanning without redundancy scheme and ii) floorplanning with redundancy scheme. The percentage of unused TSVs is based on the maximum needed TSVs for each benchmark as reported in the table.

Table 23. Comparison of number of unused TSVs between 3D floorplanning without redundancy and 3D floorplanning with redundancy. Results evaluated on final floorplan. No. of device layers = 4, Cost function = CF2, TSV Specifications: D = 3µm, H = 30 µm, TSV Pitch = 6 µm.

Test cases	Total Estimated TSVs [34]	Total Assigned TSVs		Total Spare TSVs		Total unused TSVs		% unused TSVs	
		Case I	Case II	Case I	Case II	Case I	Case II	Case I	Case II
N100	924	658	649	0	226	266	49	28.7	5.3
N200	2040	1295	1299	0	612	743	129	36.4	6.3
N300	2640	1762	1753	0	793	880	94	33.3	3.5

In case I, the floorplan obtained without TSV redundancy scheme has an average of 33% unused TSVs for all benchmarks. In contrast, the proposed redundancy scheme significantly reduces the number of unused TSVs on the layout to average 5% for all benchmarks. Further analysis on the number of assigned TSVs on individual device layers is reported in Figure 46. It shows that the implementation of the TSV redundancy scheme does not impact the number of assigned TSVs during nets-to-TSVs assignment in any significant way, when compared with case I.

6.5.2 Potential Impact on TSV Capacitance

We may recall from Section 3.5.7, that the total TSV capacitance contributed by a cluster depends on the cluster capacity. The total cluster capacitance is computed as the product of cluster capacity and the maximum (worst-case) TSV capacitance depending on the cluster size [Refer 3.3.2]. Due to the implementation of TSV redundancy in each individual cluster, some of the TSVs in each cluster get allocated as spare TSVs, therefore, reducing the effective total capacitance of that cluster. Figure 47 shows the comparison between the total capacitance contributed by individual cluster sizes, between two cases – i) without applied redundancy and ii) with applied redundancy. The total cluster capacitance in Case II significantly reduces by average 43% for all cluster capacities, as compared to case I. Furthermore, with reduction in capacitance increases with increase in cluster size. This is due to the fact that the number of spare TSVs allocated rises with cluster capacity.

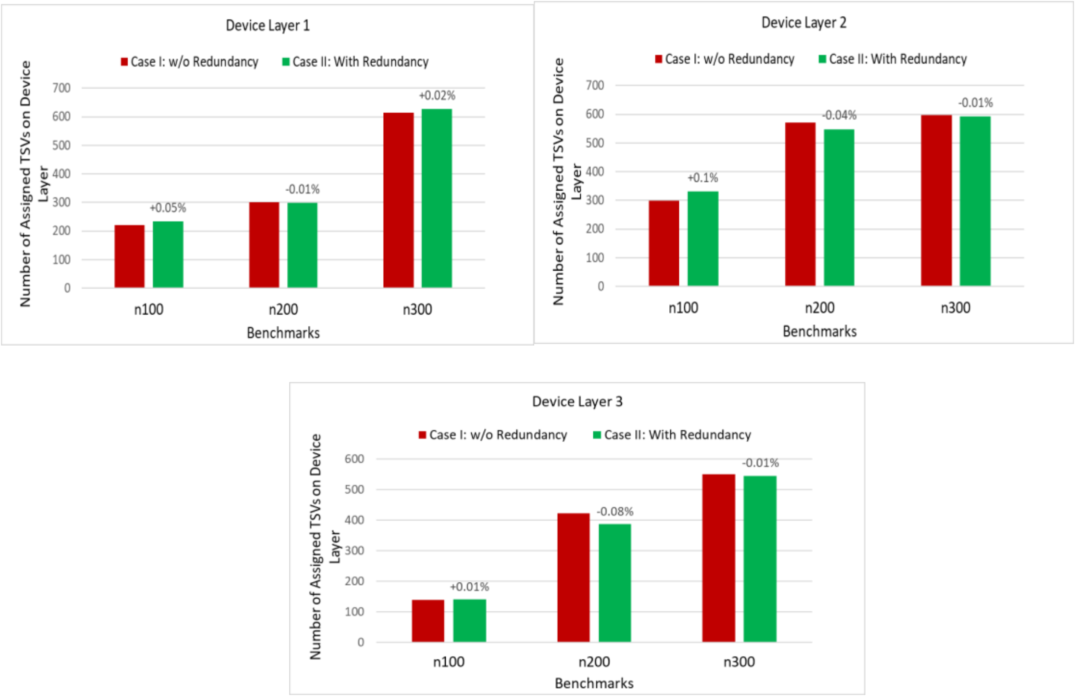


Figure 49. Comparison of assigned TSVs on individual device layers between two cases: i) 3D floorplanning without redundancy and ii) 3D floorplanning with redundancy. Note: Device Layer 3 corresponds to the topmost active layer on the 3D chip.

It is important to stress here, that within the scope of this work, we consider all TSVs in a cluster have the same capacitance as the worst-case TSV experiencing maximum coupling from neighboring TSVs. Under this assumption, we do not consider the specific position of TSVs within a cluster, whether they are situated in the middle or peripheral rows, while allocating spare TSVs in it. This implies that even if the worst-case TSV in the middle of the cluster is allocated as spare TSV, the individual TSV capacitance will still be equal to the capacitance of the worst-case TSV in the cluster. This assumption helps in the ease of estimation of delay and power during nets-to-TSVs assignment.

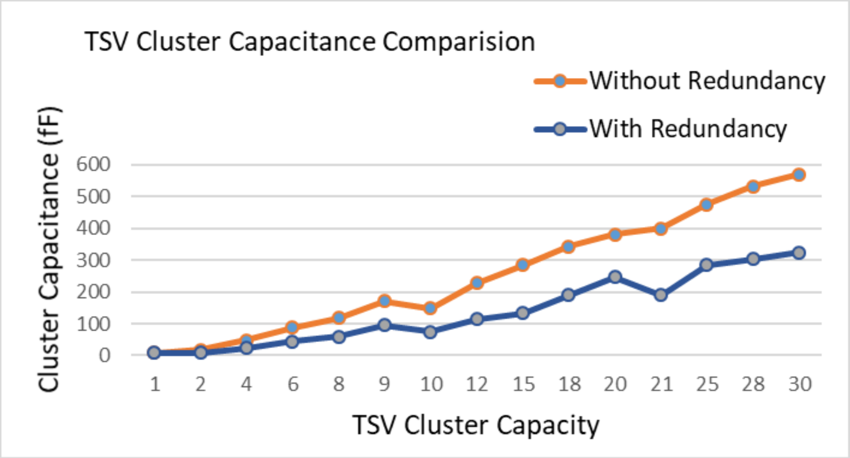


Figure 50. Comparison of total capacitance contributed by each cluster size with and without applied TSV redundancy. Increase in spare TSVs in a cluster helps contribute to reduction in cluster capacitance.

6.5.3 Impact on Delay and Power Evaluation

Table 24 shows the comparison of area, delay between delay-aware 3D floorplanning without TSV redundancy and with TSV redundancy. With the allocation of spare TSVs during redundancy, the delay improves by average 6% for all benchmarks, as compared to delay obtained without applying TSV redundancy. This is attributed to the significant reduction in capacitance contribution of individual TSV clusters as discussed in Section 6.5.2.

Another interesting phenomenon which contributes to reduction in delay in individual nets, is observed as the floorplan proceeds to convergence. It may be recalled that towards the end of iterations, the population of TSVs located in large-sized TSV clusters is much higher than small or medium-sized clusters. Hence, the probability that a net gets assigned to a TSV from a large-sized cluster goes on increasing as the floorplan converges to the final solution. Hence, in case I, without the applied TSV redundancy, more large-sized clusters on the layout will be used for net assignment, than the smaller clusters. However, with TSV redundancy, when a large sized cluster has all its functional TSVs assigned, the assignment of subsequent nets shifts to relatively smaller-sized clusters. This helps in

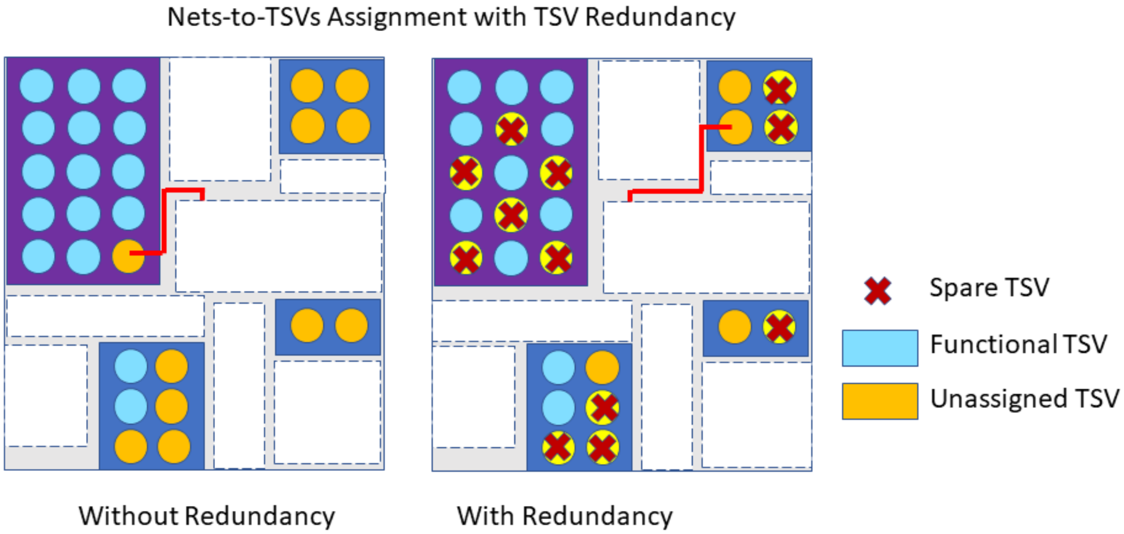


Figure 51. Illustration of differences in nets-to-TSVs assignment with and without application of TSV redundancy scheme to individual clusters.

incurring less TSV capacitance for the assigned net. This phenomenon is depicted in Fig. 51. As shown in Table 24, the impact on area due to TSV redundancy is negligible.

To further study the true impact of reduction in TSV capacitance due to TSV redundancy, we compare between the two cases, the evaluated interconnect power, which varies directly with TSV capacitance. Figure 49 shows that implementation of TSV redundancy aids insignificant 21% reduction in total interconnect power as compared to floorplanning without TSV redundancy.

Table 24. Comparison of floorplanning metrics: area and delay between two cases- with redundancy and without redundancy scheme. No. of device layers = 4, Cost function = CF2, TSV Specifications: D = 3µm, H = 30 µm, TSV Pitch = 6 µm.

Benchmark	Area (mm^2)		%diff	Delay (ps)		
	Without Redundancy	With Redundancy		Without Redundancy	With Redundancy	%diff
n100	0.256	0.252	-1.5	871	791	9.18
n200	0.288	0.279	-3.1	1487	1411	5.11
n300	0.456	0.451	-1.1	2497	2395	4.08
		Average	-1.90%		Average	6.12%

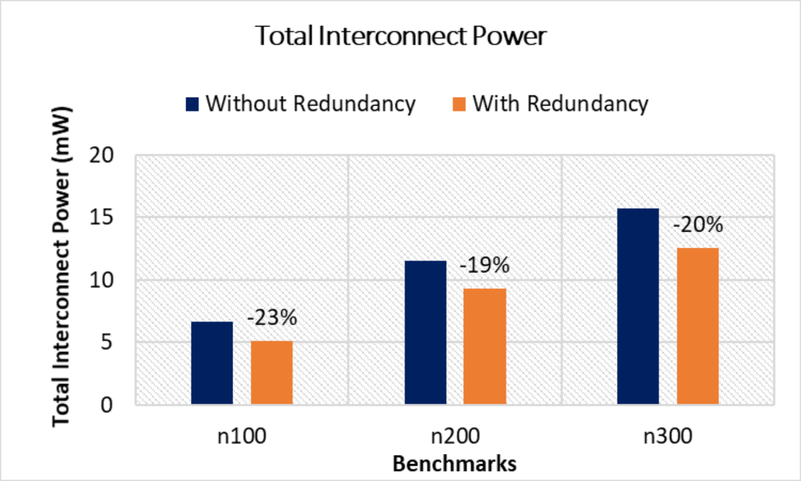


Figure 52. Comparison of total interconnect power evaluated with and without redundancy. Due to the direct dependency of power on TSV capacitance, its reduction in TSV capacitance contribution to individual nets plays a more role in power reduction.

6.5.4. Runtime Comparison

6.5.4. Runtime Comparison

A comparison of runtime between 3D floorplanning with and without TSV redundancy, shows that floorplanning with TSV redundancy slightly increases the existing runtime by 9% averaged over all benchmarks. This may be attributed to the execution of look-up table for the allocation of spare TSVs depending on cluster size for each candidate floorplan prior to nets-to-TSVs assignment. This means that a floorplan may undergo the process of spare TSV allocation in the subsequent iteration, even if the cluster sizes have not changed since previous iteration. This contributes to increase in runtime.

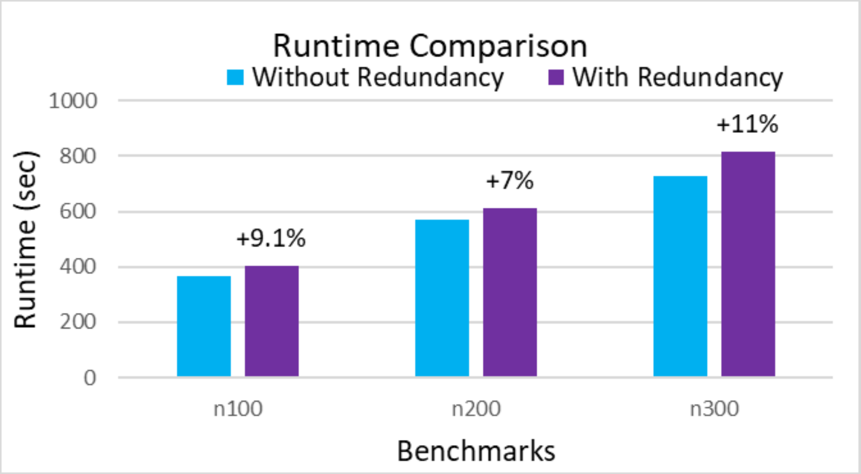


Figure 53. Comparison of runtime between two floorplanning approaches

6.6. Chapter Conclusions

A new TSV redundancy scheme is built into the existing framework of the 3D floorplanner. The required modifications in the design flow have been explained in the chapter. Based on the probabilistic model, the estimated minimum number of spare TSVs in a cluster of given size, provides the capability for full recovery of the TSV cluster in case of one failed

TSV. This further implies that a design can be fully recovered even if single TSV failure occurs in multiple TSV clusters on the layout at the same time.

The redundancy scheme increases the utilization of unused TSVs on the layout as compared to the previous version of 3D floorplanner. Most importantly, in comparison to previous approach where net assignment occurs without consideration of TSV redundancy, a signal rerouting due to a failed TSV will require routing to an alternate TSV island with available TSVs. This may critically impact delay if the rerouted signal is a timing critical signal. However, with redundancy implementation, a certain number of spare TSVs is already present within the failed cluster. Therefore, re-routing of signal within the same cluster will incur negligible wirelength overhead.

An important benefit of TSV redundancy implementation is that, the proposed approach leverages upon the performance and power benefits of dynamic TSV clustering approach. The reduction in total delay and power occur due to lowering of the total capacitance contributed by a TSV cluster, and the shifting of nets to smaller clusters for TSV assignment.

7. CONCLUSIONS AND FUTURE SCOPE OF WORK

TSV-based 3D integrated circuits have emerged as a viable alternative solution for the semiconductor industry, to keep up with the current and future demands for higher performance and functional diversification. TSVs enable vertical interconnects across stacked and thinned dies in 3D-IC designs, resulting in reduced wirelength, footprint, faster speed, improved bandwidth and lesser routing congestion. For the smooth transition of TSV-based 3D IC technology into high-volume manufacturing, it is critical that the multiple TSV-induced design complications be addressed early, as part of the layout optimization problem. Therefore, in addition to developing efficient tools for 3D layout design solution space, it is also inevitable for designers to include the impact of TSVs on layout area, interconnect delay, power and long-term reliability and yield of 3D chips.

In this thesis, we first acknowledge that performance of 3D chips is directly controlled by the quality of the generated floorplan. The core objective of this thesis is to develop an efficient methodology to improve the 3D floorplan solution quality. By generating more realistic 3D layouts, we seek to improve the accuracy of evaluation of the goodness of a 3D floorplan. A new dynamic TSV clustering algorithm is introduced, which simultaneously optimizes the sizes and positions of TSV clusters on the layout. The improved 3D floorplanning framework provides further solutions for improved estimation of buffers in 3D chips, by accurately modeling the delay of wires containing TSVs. An analytical approach is developed which prevents the excessive usage of buffers by accurately estimating the position of buffer driving a TSV. This thesis also explores the performance implications of combining nanoscale TSVs with three diverse nano-CMOS technologies yet to be considered for building of 3D ICs. A model for implementation of

TSV redundancy scheme within each TSV cluster is proposed. The developed provides full-repairability of the chip and individual TSV clusters, considering one failed TSV. This aids in increasing the fault-tolerance of TSV clusters and improving the overall reliability of the design.

7.1. Thesis Conclusions

In Chapter 3, a new dynamic TSV clustering methodology is implemented during 3D floorplanning, which results in an average 25% reduction in TSV footprint for all benchmarks, as compared to the single TSV placement approach. In addition, a novel fully non-deterministic nets-to-TSVs assignment algorithm which considers the inherent trade-off between TSV area and the TSV capacitance during net delay optimization, is included. As compared to floorplanning with fixed-sized TSV islands, the improved approach of nets-to-TSVs assignment reduces total chip area by average 7.6% and total interconnect delay by average 9%. The following papers are published/submitted related to this topic:

- S. Mohapatra, S.K.Vendra, M. Chrzanowska-Jeske, "Dynamic Through-silicon via Clustering in 3D IC Floorplanning, *(submitted to IEEE Trans. in VLSI)*
- M. A. Ahmed, S. Mohapatra, M. Chrzanowska-Jeske "TSV-and delay-aware 3D-IC floorplanning", Analog Integrated Circuits and Signal Processing, vol. 87, no. 2, pp. 235-248, March 2016.
- M. A. Ahmed, S. Mohapatra, M. Chrzanowska-Jeske, "3D Floorplanning with Netsto-TSVs Assignment", 21st IEEE Intl. Conf. on Electronics, Circuits and Systems, 26 February 2015.
- M. A. Ahmed, S. Mohapatra, M. Chrzanowska-Jeske, "Dynamic Nets-to-TSVs

Assignment in 3D Floorplanning", Intl. Symposium on Circuits and Systems, 30 July 2015.

In chapter 4, a fast and reliable method for early estimation of buffers is directly incorporated with assignment of nets to TSVs. Compared to previous approach based on fixed buffer insertion length, the current approach offers 21% more reduction in buffers and 7% reduction in delay. As compared to prior work based on fixed-sized TSV islands, the proposed method reduces estimated buffer by 7% on all device layers and 5% reduction in delay. By having a more reliable estimate of buffer insertion length the number of buffers per net is optimized. Secondly, by finding the optimal position of the driving buffer before the TSV, avoids the requirement of inserting any additional buffer around TSVs. The following papers are published/submitted related to this topic:

- S. Mohapatra, S. K. Vendra and M. Chrzanowska-Jeske, "Fast Buffer Count Estimation in 3D IC Floorplanning," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, doi: 10.1109/TCSII.2020.3007858.
- M. A. Ahmed, S. Mohapatra, M. Chrzanowska-Jeske, "Performance Optimization and Power Efficiency in 3D IC with Buffer Insertion Scheme", 29th IEEE Intl. System-on-Chip Conf. (SOCC), Sept. 6-9 2016.
- M. A. Ahmed, S. Mohapatra, M. Chrzanowska-Jeske, "Buffered Interconnects in 3D IC Layout Design", Proc. of the 18th IEEE/ACM Intl. Workshop on System level Interconnect Prediction (SLIP), 24 November 2016.

In chapter 5, the impact of three diverse nano-CMOS technology nodes, yet to be realized for future 3D ICs - 45nm, 32nm and 22nm, on early estimation of delay, buffers and dynamic power. For each technology node, we consider two different TSV diameters - 3.0µm and 0.5µm, to gain more insight into the impact of nanoscale TSVs on performance and power of future 3D ICs. With technology scaling, the total delay increases by 52% for the largest TSV size 3µm, while it rises by 48% for 0.5µm, primarily due to increase in wire resistance and rise in coupling between wire and TSVs. With the scaling of TSV diameter from 3.0 µm to 0.5 µm, the rise in delay with technology also lessens, as the TSV delay contribution relative to wire delay steadily decreases. . For 3µm TSV, the power reduces on average by 37% scaling from 45 nm to 22 nm. With TSV scaling to 0.5µm, the power reduction rises to 53%. Additionally, this work explores five different alternative TSV fill materials. Their potential effect on the critical distance between consecutive buffers (buffer insertion length) is investigated. BIL increases by 78 µm per unit ohm increase in TSV resistance for 1-TSV net, and by 120 µm for 3-TSV nets. Compared to the base case of Cu-TSV, BIL increases by 17% for SWCNT bundle TSV, 13% with Ni-TSV, 7% with W-TSV and only ~2% increase with Cu-CNT composite. The almost comparable BIL estimate and conductivity for Cu and Cu-CNT composite, indicates the promising future applications of Cu-CNT, as a direct alternative to Cu TSV, especially in the high frequency RF range.

The following work is published related to this topic:

- S. Mohapatra, S. K. Vendra and M. Chrzanowska-Jeske, "Through Silicon Via-Aware Layout Design and Power Estimation in Sub-45 Nanometer 3D CMOS IC Technologies," 2018 IEEE 13th Nanotechnology Materials and Devices Conference (NMDC), Portland, OR, USA, 2018, pp. 1-4.

In Chapter 6, a new TSV redundancy scheme is built into the existing framework of the 3D floorplanner. The required modifications in the design flow have been explained in the chapter. Based on the probabilistic model, the estimated minimum number of spare TSVs in a cluster of given size, provides the capability for full recovery of the TSV cluster in case of one failed TSV. This further implies that a design can be fully recovered even if single TSV failure occurs in multiple TSV clusters on the layout at the same time.

7.2. Future Work

This work addresses some of the most practically relevant design overheads and challenges introduced in 3D ICs due to usage of TSVs. The present work improves upon the original 3D floorplanner developed by Wang [33], Nain [35] and Ahmed [47]. Some of the possible future works that can further aid in developing the efficiency of the current version of the 3D floorplanning tool

- The method for early estimation of buffers in 3D ICs can be further improved by considering the variation of wire resistivity with temperature. Due to large-sized TSV capacitance, power density and temperature hot spots are major challenges in 3D ICs. The inclusion of these factors will further lead to more realistic estimation of delay and power in buffered 3D interconnects, especially in future nanometer technologies.
- A method for improving the estimation of maximum needed TSVs in the design can be included for minimizing the number of unused TSVs in the final layout.
- Methods to improve the runtime efficiency of the nets-to-TSVs assignment process can be included.

REFERENCES

[1] K. Banerjee, S. Souri, P. Kapur, and K. C. Saraswat, “3-D ICS: A novel chip design for improving deep-submicrometer interconnect performance and system-on-chip integration,” Proc. IEEE, vol. 89, no. 5, pp. 602–633, May 2001.

[2] W. Arden, M. Brillouët, P. Cogez, M. Graef, B. Huizing, and R. Mahnkopf. “More-than-Moore” White Paper. Tech. rep. ITRS, 2010.

[3] Shekhar Borkar. 2011. 3D integration for energy efficient system design. In Proceedings of the 48th Design Automation Conference (DAC '11). ACM, New York, NY, USA, 214-219.

[4] International Technology Roadmap for Semiconductors, “Semiconductor Industry Association,” 2015. Available: http://www.itrs2.net

[5] M. T. Bohr and I. A. Young, “CMOS Scaling Trends and Beyond,” IEEE Micro, vol. 37, no. 6, pp. 20–29, Nov. 2017.

[6] S. K. Lim, “Research needs for TSV-based 3D IC architectural floorplanning,” J. lnf. Commun. Converg. Eng., vol. 12, no. 1, pp. 46–52, 2014.

[7] Wei-ting Jonas Chan, Andrew B. Kahng, Jiajia Li, "Revisiting 3DIC Benefit with Multiple Tiers", System Level Interconnect Prediction (SLIP) 2016 ACM/IEEE International Workshop on, pp. 1-8, 2016.

[8] S. S. Iyer, “Heterogeneous Integration for Performance and Scaling”, IEEE Trans. Compon., Packag., Manuf. Technol., vol. 6, pp. 973-983, 2016.

[9] J. Knechtel, J. Lienig, “Physical Design Automation for 3D Chip Stacks: Challenges and Solutions, Proc. of Intl. Symposium on Physical Design, Pages 3-10, 2016”, IEEE

Trans. On Computer-Aided Des. Of Integr. Cir. And Sys. vol.31, no.2, pp. 228-241, Feb 2012.

[10] S. Das, A. Chandrakasan, and R. Reif, “Design tools for 3-D integrated circuits,” in Proc. of ASP-Design Automation Conf., pp. 53–56, Jan. 2003.

[11] H. Zhu, M. Zhang, Y. He and Y. Huang, "Floorplanning for 3D-IC with Through-Silicon via co-design using simulated annealing," 2018 IEEE (EMC/APEMC), Singapore, 2018, pp. 550-553.

[12] Sait, S., Oughali, F., & Al-AsliCenter, M. (2016). Design partitioning and layer assignment for 3D integrated circuits using tabu search and simulated annealing. Journal of Applied Research and Technology, 14(1)

[13] K. Kiyota and K. Fujiyoshi, "Simulated annealing search through general structure floorplans using sequence-pair," *2000 IEEE International Symposium on Circuits and Systems (ISCAS)*, Geneva, Switzerland, 2000, pp. 77-80 vol.3.

[14] Y. Ma, S. Dong, X. Hong, Y. Cai, C-K Cheng, J. Gu, “VLSI Floorplanning with Boundary Constraints Based on Corner Block List”, *Proc. Of IEEE ASP Design Automation Conf.*, pp 509-514*,* Feb 2001.

[15] H. Murata, K. Fujiyoshi and Y. Kajitani, “VLSI module placement based on rectangle-packing by the sequence pair”, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 15, no. 12, pp. 1518-1524, Aug. 2002.

[16] J.-M. Lin, Y.-W. Chang, S.-P. Lin, “Corner sequence: A P-admissible floorplan representation with a worst case linear-time packing scheme”, IEEE Trans. on VLSI Sy, Vol. 11, no. 4, pp. 679– 686, Sept. 2003.

[17] M. Kang, Wayne W. M. Dai, “General floorplanning with L-shaped, T-shaped and Soft Blocks Based on Bounded Slicing Grid Structure”, Proc. Of IEEE ASP DesignAutomation Conf., pp. 265 – 270, 06 Aug. 2002.

[18] Y.-C. Chang, Y.-W. Chang, G.-M. Wu, and S.-W. Wu, “B*-Trees: A New Representation for Non-Slicing Floorplans” Proc. Of IEEE/ACM 37th Design Automation Conf., 2000.

[19] K. Kiyota, K. Fujiyoshi, “Simulated Annealing Search Through General Structure Floorplans Using Sequence-Pair”, IEEE Symposium on Circuits and Systems, Geneva, Switzerland, pp. 77-80, Aug. 2002.

[20] T.-C. Chen, and Y.-W. Chang, “Modern Floorplanning Based on B*-Tree and Fast Simulated Annealing”, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 25, No. 4, pp. 637-650, April 2006.

[21] J. M. Lin and Y. W. Chang, “TCG-S: Orthogonal Coupling of P*-admissible Representations for General Floorplans”, Proc. Of 39th IEEE Design Automation Conf., Aug. 2002.

[22] J.-M. Lin and Y.-W. Chang, “TCG: A Transitive Closure Graph-Based Representation for General Floorplans”, IEEE Trans. on VLSI systems, Vol. 13, No. 2, pp. 288-292, Feb. 2005

[23] B. H. Gwee, and M. H. Lim, “A GA with heuristic-based decoder for IC Floorplanning”, *The VLSI journal,* Vol. 28, No. 2, pp. 157-172, Jan. 1999.

[24] X. G. Wang, L. S. Yao, and J. R. Gan, “VLSI Floorplanning Method Based on Genetic Algorithms”, Chinese Journal of Semiconductors, pp. 330-335, 2002.

[25] P. N Guo, C. K. Cheng, T., Yoshimura, “An O-treeslicing representation of non-slicing floorplan and its applications”, Proc. of 36th ACM/IEEE Design Automation Conf.., pp.268-273, Aug. 2002.

[26] H. Murata, E. Kuh, "Sequence Pair Based Module Placement for hard/soft/preplaced Modules", Intl. Symposium on Physical Design, pp. 167-172, 1998.

[27] E. Young, C. Chu and M. Ho "Placement Constraints in Floorplan Design", IEEE Trans. on VLSI, Vol. 12, No. 7, pp. 735-745, July 2004.

[28] S. N. Adya and, I. L. Markov, "Fixed-Outline Floorplanning: Enabling Hierarchical design", IEEE Trans. on VLSI Systems, vol. 11. no. 6, pp.1120 - 1135, Dec 2003.

[29] Meng-Kai Hsu, Yao-Wen Chang, and Valeriy Balabanov. 2011. TSV-aware analytical placement for 3D IC designs. In Proceedings of the 48th Design Automation Conference (DAC '11). ACM, New York, NY, USA, 664-669.

[30] Y. Tam, E. F. Y. Young and C. Chu, "Analog Placement with Symmetry and other Placement Constraints", Intl. Conf. on Computer Aided Design, pp. 349-354, 2006.

[31] X. Liu, Y. Zhang, G. Yeap and X. Zeng, "An integrated algorithm for 3D-IC TSV assignment," *2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, New York, NY, 2011, pp. 652-657.

[32] M. Chrzanowska-Jeske, B. Wang and G. Greenwood, "Floorplanning with Performance-based Clustering", Proc. Int. Symposium on Circuits and Systems, vol. 4, pp.724-727, 2003.

[33] B. Wang, M. Chrzanowska-Jeske, and G. Greenwood, "ELF-SP-Evolutionary Algorithm for Non-Slicing Floorplans with Soft Modules", Proc. Int. Symposium on Circuits and Systems, vol.2, pp.681-684, 2002.

[34] J. Cong and G. Luo, "A multilevel analytical placement for 3D ICs," 2009 Asia and South Pacific Design Automation Conference, Yokohama, 2009, pp. 361-366.

[35] R.K. Nain, M. Chrzanowska-Jeske, “Fast Placement Aware 3-D Floorplanning using Vertical Constraints on Sequence Pairs,” IEEE Trans. On VLSI systems, vol. 19, pp. 1667-1680, Sept. 2011.

[36] P. Zhou, Y. Ma, Z. Li, R. P. Dick, L. Shang, H. Zhou, Q. Zhou, and X. Hong, “3-DSTAF: Scalable temperature and leakage aware floorplanning for 3-D integrated circuits,” in Proc. IEEE Int. Conf. Computer-Aided Design, pp.590–597, Nov. 2007.

[37] Z. Li, X. Hong, Q. Zhou, Y. Cai, J. Bian, H. H. Yang, V. Pitchumani, and C.-K. Cheng, “Hierarchical 3-D floorplanning algorithm for wirelength optimization,” IEEE Trans. On Circuits and Syst. I, vol. 53, no. 12, pp. 2637–2646, Dec. 2006.

[38] Y. Qian, C. Hao and T. Yoshimura, "3D-IC signal TSV assignment for thermal and wirelength optimization," 2017 27th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS), Thessaloniki, 2017, pp. 1-8.

[39] J. Cong, J. Wei, and Y. Zhang, “A thermal-driven floorplanning algorithm for 3-D ICs,” in Proc. IEEE Int. Conf. Computer-Aided Design, pp. 306–313 Nov. 2004.

[40] K. Athikulwongse, D. H. Kim, M. Jung and S. K. Lim, "Block-level designs of die-to-wafer bonded 3D ICs and their design quality tradeoffs," 2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC), Yokohama, 2013, pp. 687-692

[41] S. Chen and T. Yoshimura, “Multilayer floorplanning for stacked ICs: Configuration number and fixed-outline constraints,” Integr., Journal of VLSI vol. 43, no. 4, pp. 378– 388, Sep. 2010.

[42] J. Knechtel, I.L. Markov, J. Lienig, “Assembling 2-D Blocks into 3-D Chips”, IEEE Trans. On Computer-Aided Design Of Integr. Cir. And Sys. vol.31, no.2, pp. 228-241, Feb 2012.

[43] M.C. Tsai, T.C. Wang, and T.T. Hwang, “Through-silicon via planning in 3-D floorplanning,” IEEE Trans. On VLSI systems, vol. 19, no.8, pp. 1448-1457, Aug 2011.

[44] C.R Li, W.K Mak, and T.C Wang, “Fast Fixed-Outline 3-D IC Floorplanning with TSV Co-Placement,” IEEE Trans. On VLSI systems, vol. 21(3), pp. 523-532, Mar 2013.

[45] D. H. Kim K. Athikulwongse S. K. Lim "Study of through-silicon-via impact on the 3-D stacked IC layout" IEEE Trans. Very Large Scale Integr. (VLSI) Syst. vol. 21 no. 5 pp. 862-874 May 2013.

[46] J. Knechtel, E. Young, and J. Lienig, “Planning massive interconnects in 3D chips,” Trans. CAD, vol. 34, no. 11, pp. 1808–1821, 2015.

[47] M. A. Ahmed, S. Mohapatra, and M. Chrzanowska-Jeske, “TSV-and delay-aware 3D-IC floorplanning,” Analog Integr. Circuits Signal Process., vol. 87, no. 2, pp. 235–248, May 2016.

[48] A. Quiring, M. Olbrich and E. Barke, "Fast global interconnnect driven 3D floorplanning," 2015 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), Daejeon, 2015, pp. 313-318.

[49] L. Lyu and T. Yoshimura, "A force directed partitioning algorithm for 3D floorplanning," 2017 IEEE 12th International Conference on ASIC (ASICON), Guiyang, 2017, pp. 718-721.

[50] D. Saha and S. Sur-Kolay, "Multi-objective Optimization of Placement and Assignment of TSVs in 3D ICs," 2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID), Hyderabad, 2017, pp. 372-377.

[51] Johann Knechtel, Jens Lienig, and Ibrahim (Abe) M. Elfadel. 2017. Multi-Objective 3D Floorplanning with Integrated Voltage Assignment. ACM Trans. Des. Autom. Electron. Syst. 23, 2, Article 22 (November 2017), 27 pages.

[52] Lin, Jai-Ming and Jung-An Yang. “Routability-Driven TSV-Aware Floorplanning Methodology for Fixed-Outline 3-D ICs.” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 36 (2017): 1856-1868.

[53] H. Kalargaris, Y. Chen and V. F. Pavlidis, "STA compatible backend design flow for TSV-based 3-D ICs," 2017 18th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, 2017, pp. 186-190.

[54] G. Katti, M. Stucchi, K. D. Meyer, and W. Dehaene, “Electrical Modeling and Characterization of Through Silicon via for Three-Dimensional ICs,” in IEEE Trans. on Electron Devices, no. 1, Jan. 2010, pp. 256–262.

[55] M. A. Ahmed and M. Chrzanowska-Jeske, “TSV Capacitance Aware 3D Floorplanning,” 3D System Integration Conf., 09 January 2014.

[56] D.H. Kim, S Mukhopadhyay, S.K. Lim, “Fast and Accurate Analytical Modeling of Through-Silicon-Via Capacitive Coupling,” IEEE Trans. On Advanced Packaging 2011. I. Savidis and E. G. Friedman, “Closed-Form Expressions of 3D Via Resistance, Inductance, and Capacitance,” in IEEE Trans. On Electron Devices, vol 56, no. 9, Sept 2009, pp. 1873-1881.

[57] T. Sakurai and K. Tamaru, “Simple Formulas for Two- and Three-Dimensional Capacitances,” in IEEE Trans. On Electro Dev., vol. 30, no. 2, Feb 1983, pp. 183- 185.

[58] F.C. Wu, S.C. Wong, P.S .Liu, D.L.Yu, F. Lin, “Empirical Models for wiring Capacitances in VLSI,” In Proc. IEEE Int. Sym. On Circuits and Systems, 1996.

[59] D. Khalil, Y. Ismail, M. Khellah, T. Karnik, V. De, “Analytical Model for propagation delay of Through Silicon Vias,” in Proc. Int. Symp. On Quality Electronic Design, March 2008, pp. 553-556.

[60] D.H. Kim, S. Mukhopadhyay, and S. K. Lim. "TSV-aware interconnect distribution models for prediction of delay and power consumption of 3-d stacked ICs," Computer-Aided Design of Integrated Circuits and Systems, IEEE Trans. on 33.9 (2014): 1384-1395.

[61] M. A. Ahmed and M. Chrzanowska-Jeske, "Delay and power optimization with TSV-aware 3D floorplanning," Fifteenth International Symposium on Quality Electronic Design, Santa Clara, CA, 2014, pp. 189-196.

[62] M.A. Ahmed, S. Mohapatra, and M. Chrzanowska-Jeske, “3D floorplanning with nets-to-TSVs assignment," 21st IEEE Intl. Conf. on Electronic, Circuits and Systems, pp.578,581, 7-10 Dec. 2014.

[63] M.A. Ahmed, S. Mohapatra and M. Chrzanowska-Jeske, "Dynamic nets-to-TSVs assignment in 3D floorplanning," 2015 IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, 2015, pp. 1870-1873.

[64] W. C. Elmore, “The transient response of damped linear network with particular regard to wideband amplifiers”, Journal of Applied Physics, 19, 55–63, 1948.

[65] T. Bandyopadhyay, R. Chatterjee, D. Chung, M. Swaminathan, and R. Tummala, “Electrical Modeling of Through Silicon and Package Vias,” in Proc. IEEE Int. Conf. on 3D System Integration, Sept. 2009.

[66] C. Xu et al., “Compact AC Modeling and Performance Analysis of Through-Silicon Vias in 3-D ICs,” IEEE Trans. on Electron Devices, vol. 57, no. 12, pp. 3405–3417, 2010.

[67] G. Neela, J. Draper, “Congestion-aware optimal techniques for assigning inter-tier signals to 3D-vias in a 3DIC”, Intl. Conf. on 3D Systems Integration, Aug 31 – Sept 2 2015.

[68] Intel Technology Journal, “Intel’s 45nm CMOS Technology”, vol. 12, no. 02, June 17, 2008.

[69] S. Natarajan et al., "A 32nm logic technology featuring 2nd-generation high-k + metal-gate transistors, enhanced channel strain and 0.171μm2 SRAM cell size in a 291Mb array," 2008 IEEE International Electron Devices Meeting, San Francisco, CA, 2008, pp. 1-3.

[70] J. P. Gambino, "Copper interconnect technology for the 22 nm node," Proceedings of 2011 International Symposium on VLSI Technology, Systems and Applications, Hsinchu, 2011, pp. 1-2.

[71] D. H. Kim, S. Kim and S. K. Lim, "Impact of nano-scale through-silicon vias on the quality of today and future 3D IC designs," Internatl. Workshop on System Level Interconnect Prediction, San Diego, CA, 2011, pp. 1-8. (SLIP), Jun. 2011, pp.1–8.

[72] D. H. Kim and S. K. Lim, "Design Quality Trade-Off Studies for 3-D ICs Built With Sub-Micron TSVs and Future Devices," in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 2, no. 2, pp. 240-248, June 2012.

[73] S. Mohapatra, S. K. Vendra and M. Chrzanowska-Jeske, "Through Silicon Via-Aware Layout Design and Power Estimation in Sub-45 Nanometer 3D CMOS IC Technologies," 2018 IEEE 13th Nanotechnology Materials and Devices Conference (NMDC), Portland, OR, USA, 2018, pp. 1-4.

[74] S. Dong, H. Bai, X. Hong, and S. Goto. "Buffer Planning for 3D ICs," In Proc IEEE Int. Symp. on Circuits and Systems, pages 1735–1738, 2009.

[75] X. He, S. Dong, Y. Ma, and X. Hong, "Simultaneous buffer and interlayer via planning for 3D floorplanning," in Proc. of the 10th Internl. Sympo. on Quality Electronic Design. pp. 740–745, March 2009.

[76] Y. Lee, and S. K. Lim. "Fast delay estimation with buffer insertion for through silicon- via-based 3D interconnects," Quality Electronic Design (ISQED), 2012 13th Intl. Symposium on. IEEE, 2012.

[77] M. A. Ahmed, S. Mohapatra and M. Chrzanowska-Jeske, "Buffered interconnects in 3D IC layout design," 2016 ACM/IEEE Intl. Workshop on System Level Interconnect Prediction (SLIP), Austin, TX, pp. 1-8.

[78] D. H. Kim and S. K. Lim, "Through-Silicon-Via-Aware Delay and Power Prediction Model for Buffered Interconnects in 3D ICs," in ACM/IEEE International Workshop on System Level Interconnect Prediction, 2010, pp. 25–32.

[79] M. A. Ahmed, S. Mohapatra and M. Chrzanowska-Jeske, "Performance optimization and power efficiency in 3D IC with buffer insertion scheme," 2016 29th

IEEE International System-on-Chip Conference (SOCC), Seattle, WA, 2016, pp. 229-234.

[80] M.A. Ahmed, M. Chrzanowska-Jeske, “TSVs in Early Layout Design Exploration for 3D ICs”, IEEE 5th Latin Amer Symp. On Circ. and Syst, (LASCAS), pp.25-28, Feb 2012.

[81] J. Yang, K. Athikulwongse, Y. Lee, S. K. Lim and D. Z. Pan, "TSV stress aware timing analysis with applications to 3D-IC layout optimization," Design Automation Conference, Anaheim, CA, 2010, pp. 803-806.

[82] M. Jung, J. Mitra, D. Pan, S. Lim, “TSV stress-aware full-chip mechanical reliability analysis and optimization for 3D IC”, Communications of the ACM, vol.57 no.1, January 2014.

[83] A.-C. Hsieh, T. Hwang, M.-T. Chang, M.-H. Tsai C.-M. Tseng, H.-C. Li, “TSV redundancy: Architecture and design issues in 3D IC,” Design, Automation & Test in Europe Conf. & Exhibition, pp. 166–171, April 2010.

[84] Q. Xu, S. Chen, X. Xu and B. Yu, "Clustered Fault Tolerance TSV Planning for 3-D Integrated Circuits," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 8, pp. 1287-1300, Aug. 2017.

[85] S. Chen, Q. Xu and B. Yu, "Adaptive 3D-IC TSV Fault Tolerance Structure Generation," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 5, pp. 949-960, May 2019.

[86] Shen, W., Chen, K. Three-Dimensional Integrated Circuit (3D IC) Key Technology: Through-Silicon Via (TSV). Nanoscale Res Lett 12, 56 (2017) doi:10.1186/s11671-017-1831-4

[87] Rajeev K. Nain, SP-based Floorplanning software and its source code for PhD thesis, Electrical and Computer Engineering Department, Portland State University, 2011.

[88] Mohammad A. Ahmed, Co-placement of TSVs with Circuit Blocks Floorplanning software and its source code for Ph.D Thesis, Electrical & Computer Engineering, Portland State University, 2017.

[89] P. Saxena, N. Menezes, P. Cocchini, and D. A. Kirkpatrick. Repeater scaling and its impact on CAD. IEEE Transactions on Computer-Aided Design, 23(4):451–463, April 2004.

[90] C. Subramaniam et al., “One hundred fold increase in current carrying capacity in a carbon nanotube–copper composite,” Nature Commun.,vol. 4, p. 2202, Jul. 2013.

[91] Chuan Xu, Hong Li, R. Suaya and K. Banerjee, "Compact AC modeling and analysis of Cu, W, and CNT based through-silicon vias (TSVs) in 3-D ICs," 2009 IEEE Internl. Electron Devices Meeting (IEDM), Baltimore, MD, 2009, pp. 1-4.

[92] M.A Ahmed, M.Chrzanowska-Jeske, “Delay and Power Optimization with TSV-Aware 3D Floorplanning,” Proc. of the IEEE Internl Symp. of Qual. Elect Design, March, 2014.

[93] F. Santagata et al., "Fully back-end TSV process by Cu electro-less plating for 3D smart sensor systems," SENSORS, 2012 IEEE, Taipei, 2012, pp.1-4.

[94] G. Parès et al., "Through Silicon Via technology using tungsten metallization," 2011 IEEE International Conference on IC Des & Technology, Kaohsiung, 2011, pp. 1-4.

[95] Iniewski, K. (Ed.), “Develo of 3D Chip Integ Tech.” in “Nano-semiconductors:Dev and Tech”, 1st ed, Boca Raton: CRC Press, 2011, pp. 185-188.

[96] Vaisband, Boris, et al. "Electrical and Thermal Models of CNT TSV and Graphite Interface." IEEE Transactions on Electron Devices 65.5 (2018): 1880-1886.

[97] A. Gupta, B. C. Kim, S. Kannan, S. S. Evana and L. Li, "Analysis of CNT based 3D TSV for emerging RF applications," 2011 IEEE 61st Electronic Components and Technology Conference (ECTC), Lake Buena Vista, FL, 2011, pp. 2056-2059.

[98] Y. Feng and S. Burkett, “Fabrication and electrical performance of through silicon via interconnects filled with a copper/carbon nanotube composite,” Journ of Vac Science & Technology B, vol. 33, no. 2, 2015.

[99] M. Rao, “Electrical modeling of copper/carbon nanotubes for 3d integration,” in Nanotechnology (IEEE-NANO), 2016 16th IEEE Conf on, Aug 2016, pp. 763–766.

[100] L. Jiang, et al., "Yield enhancement for 3D-stacked memory by redundancy sharing across dies," ICCAD, pp.230-234, 2010.

[101] R. Nain et al., "Yield improvement of 3D ICs in the presence of defects in through signal vias," ISQED, pp.598-605, 2010.

[102] I. Loi, et al., "A low-overhead fault tolerance scheme for TSV-based 3-D network on chip links", ICCAD, pp.598 - 602, 2008.

[103] Y.-G. Chen, W.-Y. Wen, Y. Shi, W.-K. Hon, and S.-C. Chang, “Novel spare TSV deployment for 3-D ICs considering yield and timing constraints,” IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 34, no. 4, pp. 577–588, 2015.

[104] S. Wang, M. B. Tahoori, and K. Chakrabarty, “Defect clustering aware spare-TSV allocation for 3D ICs,” in Proc. IEEE/ACM Int.Conf. Comput.-Aided Design, Austin, TX, USA, Nov. 2015, pp. 307–314.

[105] S. Mohapatra, S. K. Vendra and M. Chrzanowska-Jeske, "Fast Buffer Count Estimation in 3D IC Floorplanning," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, doi: 10.1109/TCSII.2020.3007858.

[106] Lennart Bamberg, Amir Najafi, Alberto García-Ortiz, Edge effects on the TSV array capacitances and their performance influence, *Integrat*, Vol. 61, 2018, pp 1-10,

[107] N. Chiodarelli et al., J. Electrochem. Soc., vol. 157, p. K211, 2010.

[108] A. Stillmaker and B. Baas, “Scaling equations for the accurate prediction of CMOS device performance from 180 nm to 7 nm,” Integration, vol. 58, pp. 74–81, Jun. 2017.