

# UC Irvine

## UC Irvine Electronic Theses and Dissertations

### Title

Resilient 3D Network-on-Chip Design and Analysis

### Permalink

<https://escholarship.org/uc/item/5615x1tc>

### Author

Yaghini, Pooria M.

### Publication Date

2016

### Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,  
IRVINE

Resilient 3D Network-on-Chip Design and Analysis

DISSERTATION

submitted in partial satisfaction of the requirements  
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical and Computer Engineering

by

Pooria Mohammadi Yaghini

Dissertation Committee:  
Professor Nader Bagherzadeh, Chair  
Professor Jean-Luc Gaudiot  
Professor Alex Nicolau

2016



Chapter 2 © 2015 IEEE

Chapter 3 © 2015 ACM

Chapter 4 © 2015 IEEE

Chapter 5 © 2015 IEEE

Chapter 6 © 2015 IEEE

Chapter 7 © 2016 IEEE

All other materials © 2016 Pooria Mohammadi Yaghini

# DEDICATION

To

my dearest wife Maryam,  
for her true love and support

# TABLE OF CONTENTS

	Page
<b>LIST OF FIGURES</b>	<b>vi</b>
<b>LIST OF TABLES</b>	<b>viii</b>
<b>ACKNOWLEDGMENTS</b>	<b>ix</b>
<b>CURRICULUM VITAE</b>	<b>x</b>
<b>ABSTRACT OF THE DISSERTATION</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Basics of Resilience . . . . .	7
1.2 Basics of NoC . . . . .	10
1.3 Dissertation Contributions . . . . .	12
1.4 Dissertation Organization . . . . .	14
<b>I Fundamentals, Analysis and Modeling</b>	<b>17</b>
<b>2 Three-dimensional Networks-on-Chip Reliability Concerns</b>	<b>18</b>
2.1 Motivation . . . . .	19
2.2 3D-NoC Architecture . . . . .	21
2.3 Reliability Concerns in Three-dimensional Networks-on-Chip (3D-NoC)	23
2.3.1 Physical-level Potential Faults . . . . .	25
2.3.2 Logic-level Fault Models . . . . .	31
2.4 Fault Effects on 3D-NoC . . . . .	33
2.5 Failure Analysis of 3D-NoC . . . . .	35
2.5.1 TSV Issues . . . . .	36
2.5.2 Thermal Concern . . . . .	43
2.5.3 SEE Impacts . . . . .	47
2.6 Summary . . . . .	48

<b>3</b>	<b>Formal Reliability Analysis of 3D-NoC</b>	<b>49</b>
3.1	Introduction . . . . .	49
3.2	Reliability Analysis . . . . .	51
3.2.1	Source-Destination Status . . . . .	52
3.2.2	TSV Status . . . . .	53
3.2.3	Reliability Analysis . . . . .	54
3.3	Summary . . . . .	65
<b>4</b>	<b>System-level TSV Coupling Fault Model</b>	<b>66</b>
4.1	Introduction . . . . .	66
4.2	TTCC Elaboration . . . . .	68
4.2.1	TTCC Circuit-level Modeling . . . . .	68
4.2.2	TTCC Effect on TV . . . . .	73
4.3	TSV Coupling Fault Model . . . . .	75
4.4	Case Study: Diagnosing TSV Coupling at Runtime in 3D-NoC . . . . .	79
4.4.1	System Configuration . . . . .	79
4.4.2	Fault model accuracy . . . . .	80
4.4.3	TSV Coupling Fault Characterization . . . . .	81
4.5	Summary . . . . .	83
	<b>II TSV Coupling Mitigation</b>	<b>85</b>
<b>5</b>	<b>Inductive TSV Coupling Mitigation</b>	<b>86</b>
5.1	Introduction . . . . .	86
5.2	Inductive TSV-to-TSV Coupling . . . . .	88
5.2.1	Inductive Coupling Characteristics . . . . .	88
5.2.2	Problem Definition . . . . .	92
5.2.3	Coding Scheme . . . . .	94
5.2.4	System Model . . . . .	95
5.3	Baseline Coding Algorithm . . . . .	96
5.3.1	Effect of bit inversion on TSV current . . . . .	96
5.3.2	Reducing the sum of neighbor currents by inversion . . . . .	97
5.3.3	An Example of Baseline Algorithm . . . . .	101
5.4	Baseline Algorithm Evaluation . . . . .	104
5.4.1	Evaluation Metrics . . . . .	106
5.4.2	Scalability of Baseline Algorithm . . . . .	107
5.5	Enhanced Coding Algorithm . . . . .	111
5.5.1	Partitioning Approach . . . . .	112
5.5.2	Enhanced Algorithm Evaluation . . . . .	113
5.5.3	Hardware Synthesis Results . . . . .	115

5.6	Summary . . . . .	117
<b>6</b>	<b>Capacitive TSV Coupling Mitigation</b>	<b>118</b>
6.1	Introduction . . . . .	118
6.2	Proposed coding approaches . . . . .	119
6.2.1	Baseline TCMA . . . . .	120
6.2.2	Enhanced TCMA . . . . .	120
6.3	TCMA Elaboration and Evaluation . . . . .	127
6.4	Summary . . . . .	131
<b>7</b>	<b>Asynchronous Architecture to Avoid TSV Coupling</b>	<b>133</b>
7.1	Introduction . . . . .	133
7.2	CTCA Technique . . . . .	134
7.2.1	Dual-rail Coding Concept . . . . .	135
7.2.2	Parasitic capacitance elimination with proposed coding . . . . .	136
7.2.3	Supporting Architecture for Dual-rail Coding . . . . .	138
7.3	Evaluation of CTCA Methods . . . . .	142
7.4	Summary . . . . .	145
<b>8</b>	<b>Conclusions and Future Roadmap</b>	<b>146</b>
8.1	Put it All Together . . . . .	146
8.2	Summary of Contributions . . . . .	147
8.3	Future Work . . . . .	150
8.3.1	Extending Dynamic TSV Coupling Fault Model . . . . .	150
8.3.2	Coupling Fault Avoidance or Recovery . . . . .	150
8.4	Concluding Remarks . . . . .	150
	<b>Bibliography</b>	<b>151</b>



# LIST OF FIGURES

	Page
1.1 Wire delay versus gate delay [38] . . . . .	4
1.2 Processing Memory Elements (PME), as an example of Three-dimensional (3D) IC. This architecture depicts a 3D IC with memory, CPU, GPU, and interconnection network stacked on top of each other. . . . .	5
1.3 Network-on-Chip (NoC) architecture; $5 \times 5$ mesh topology . . . . .	6
1.4 Fault means to reach dependability . . . . .	8
1.5 NoC router architecture for distributed routing . . . . .	11
2.1 3D-NoC structure . . . . .	21
2.2 Potential physical-level faults in 3D-NoC and their corresponding logic-level models . . . . .	24
2.3 Classification of faults and their effects in 3D-NoC components . . . . .	34
2.4 Current flow direction in TSV. . . . .	36
2.5 Different TSV patterns leading to coupling . . . . .	38
3.1 A sample of error function vs number of iterations for an $8 \times 8 \times 8$ router network supported by and $8 \times 8 \times 7$ TSV network, $n_s = N_r/4$ , with dimension order routing. . . . .	61
3.2 Variation of $P_{T S}(t s)$ with number of active sources and TSVs . . . . .	62
3.3 Variation of system failure probability with injection rate and TSV failure probability . . . . .	64
4.1 Current and TSV-to-TSV Capacitive Coupling (TTCC) matrices in a $3 \times 3$ mesh of TSVs . . . . .	69
4.2 Measure the induced capacitive coupling on each TSV based on physical parameters . . . . .	70
4.3 Characterizing TSV coupling against various parameters . . . . .	72
4.4 Probability of TV in PARSEC benchmark workloads under different conditions . . . . .	73
4.5 3D IC structure with the proposed capacitive coupling fault model . . .	75
4.6 Fault model usage demonstration in 3D-IC . . . . .	77

4.7	Inaccuracy introduced by random fault models . . . . .	81
4.8	Fault model usage demonstration in 3D-NoC running PARSEC benchmark . . . . .	82
4.9	Fault density map . . . . .	83
5.1	3D-NoC vault, vertically interconnected by TSV bus in 3D integration technology . . . . .	87
5.2	Inductive coupling SPICE simulation results . . . . .	90
5.3	Evaluating the efficiency of baseline algorithm for an $8 \times 8$ TSV bus with PARSEC data traffic. For each workload the left bar represents the uncoded and the right bar shows the coded approach results. . . . .	104
5.4	Evaluating the efficiency of baseline algorithm for an $8 \times 8$ TSV bus with random data traffic. . . . .	105
5.5	ICM gain versus number of columns . . . . .	106
5.6	ICM gain versus number of rows . . . . .	108
5.7	ICM gain and corresponding information redundancy for the same number of bits in a bus with column or row growth . . . . .	109
5.8	Probability of row inversion versus number of cells in a row . . . . .	112
5.9	ICM gain for the same bus size with different partitions (P) vs row ( $N_R$ ) growth . . . . .	114
5.10	Information redundancy overhead rate for the same bus size with different partitions (P) vs row ( $N_R$ ) growth . . . . .	114
5.11	Encoder silicon area . . . . .	116
5.12	Proposed encoder power consumption versus TSV bus size . . . . .	116
6.1	Probability of bad configuration occurrence . . . . .	121
6.2	Potential configurations to generate 7C and 8C parasitic capacitance . . . . .	123
6.3	An example of baseline algorithm issues . . . . .	127
6.4	Number of 7C/8C for random data bit patterns in small mesh of TSVs . . . . .	129
6.5	7C and 8C parasitic capacitance for random and PARSEC applications data with/without TCMA . . . . .	130
7.1	Dual-rail encoding . . . . .	135
7.2	TSVs' current flow in dual-rail coding . . . . .	137
7.3	Dual-rail coding implemented using GALS architecture . . . . .	140
7.4	GALS wrapper circuit, (a) sync to async, (b) async to sync . . . . .	141
7.5	GALS wrapper STGs, (a) sync to async, (b) async to sync . . . . .	142
7.6	Capacitive coupling percentages before and after employing CTCA for both random and PARSEC workloads . . . . .	144

# LIST OF TABLES

	Page
2.1 Physical faults and their corresponding logic-level fault models . . . . .	32
2.2 TSV-to-TSV inductive coupling categorization . . . . .	40
2.3 TSV-to-TSV capacitive coupling categorization . . . . .	41
4.1 Simulation configuration settings . . . . .	74
4.2 System configuration parameters . . . . .	79
5.1 Simulation configurations in Figure 5.2 . . . . .	91
5.2 Reducing the sum of neighbor currents by inverting vertical neighbors .	99
5.3 Proposed encoder latency versus TSV bus size ( <i>ps</i> ) . . . . .	117
6.1 Current flow of TSVs before and after encoding . . . . .	122
6.2 Hardware synthesise results . . . . .	130
7.1 Circuit-level model results of CTCA . . . . .	143

# ACKNOWLEDGMENTS

First of all, I am grateful to my advisor, Professor Nader Bagherzadeh, for his guidance and patience over these years. He has always been very supportive and encouraged me throughout this research. Working with him has been an extremely valuable learning experience. Without his supervision, this dissertation would not have been possible. I also express my gratitude to Jean-Luc Gaudiot and Professor Alex Nicolau for serving on my dissertation committee. Their insightful comments and suggestions helped improve this dissertation.

Many thanks to my colleagues at Advanced Computer Architecture Group, specially Ashkan Eghbal, Abdulaziz Alhussien, Siavash S. Yazdi, and Misagh Khayambashi. Cooperation and discussion with them have been very helpful for my research. Their friendship made our laboratory an enjoyable place to work.

My deepest appreciation goes to my family, especially my parents and my wife Maryam, for their love and support. They gave me enormous courage to overcome any difficulties. I am also thankful for all of my dear friends who have helped me to go through all the challenges.

# CURRICULUM VITAE

Pooria Mohammadi Yaghini

## EDUCATION

<b>Ph.D. in Computer Systems and Software</b> University of California, Irvine	<b>May 2016</b> <i>Irvine, California</i>
<b>M.S. in Computer Engineering</b> Amirkabir University of Technology	<b>Dec. 2009</b> <i>Tehran, Iran</i>
<b>B.S. in Computer Engineering</b> Sadjad University of Technology	<b>Sep. 2006</b> <i>Mashad, Iran</i>

## EXPERIENCE

<b>Graduate Student Researcher</b> University of California, Irvine	<b>Sep. 2011–May 2016</b> <i>Irvine, California</i>
<b>Research Intern</b> STEC Corporation	<b>Jul. 2012–Sep. 2013</b> <i>San diego, California</i>
<b>Research Intern</b> HGST, a Western Digital company	<b>Oct. 2013–Apr. 2015</b> <i>San diego, California</i>

## RESEARCH INTERESTS

3D On-chip networks, Reliable system design, High performance NoC design, Computer architecture, many-core architectures, cache coherence protocols, emerging applications for many-core architectures.

## AWARDS

<b>Graduate Student Fellowship</b> University of California, Irvine	<b>2011–2012</b>
<b>Featured paper of IEEE Transaction on Computers on December 2015</b>	<b>2015</b>

## PUBLICATIONS

1. Freek Verbeek, **Pooria M. Yaghini**, Ashkan Eghbal and Nader Bagherzadeh, “Deadlock Verification of Cache Coherence Protocols and Communication Fabrics”, in Proceedings of the Design, Automation and Test in Europe (DATE) Conference, 2016.
2. **Pooria M. Yaghini**, Ashkan Eghbal, Siavash S. Yazdi, and Nader Bagherzadeh, “Accurate System-level TSV-to-TSV Capacitive Coupling Fault Model for 3D-NoC”, 9<sup>th</sup> International Symposium on Networks-on-Chip (NOCS), Vancouver, Canada, September 2015.
3. **Pooria M. Yaghini**, Ashkan Eghbal, Siavash S. Yazdi, Nader Bagherzadeh, and Michael M. Green, “Capacitive and Inductive TSV-to-TSV Resilient Approaches for 3D ICs,” *Computers, IEEE Transactions on*, vol. 65, no. 3, pp. 693-705, March 1 2016.
4. Ashkan Eghbal, **Pooria M. Yaghini**, Nader Bagherzadeh, and Misagh Khayambashi, “TSV Analytical Fault Tolerance Assessment for 3D Network-on-Chip,” *Computers, IEEE Transactions on*, vol. 64, no. 12, pp. 3591-3604, Dec. 1 2015.
5. **Pooria M. Yaghini**, Ashkan Eghbal, Misagh Khayambashi, and Nader Bagherzadeh, “Coupling Mitigation in 3-D Multiple-Stacked Devices,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 23, no. 12, pp. 2931-2944, Dec. 2015.
6. **Pooria M. Yaghini**, Ashkan Eghbal, and Nader Bagherzadeh, “On the Design of Hybrid Routing Mechanism for Mesh-based Network-on-Chip”, *Integration, the VLSI Journal Elsevier*, vol 50, pp. 183-192, June 2015.
7. Misagh Khayambashi, **Pooria M. Yaghini**, Ashkan Eghbal, and Nader Bagherzadeh, “Analytical Reliability Analysis of 3D NoC Under TSV Failure,” *ACM Journal on Emerging Technologies in Computing Systems*, vol 11, no. 4, pp. 43:1–43:16, April 2015.
8. Ashkan Eghbal, **Pooria M. Yaghini**, and Nader Bagherzadeh, “Capacitive Coupling Mitigation for TSV-based 3D ICs”, IEEE VLSI Test Symposium, (VTS 2015), pp.1-6, April 2015..
9. Ashkan Eghbal, **Pooria M. Yaghini**, Siavash S. Yazdi, and Nader Bagherzadeh, “TSV-to-TSV Inductive Coupling-aware Coding Scheme for 3D Network-on-Chip,” *Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2014 IEEE International Symposium on*, pp.92,97, 1-3 Oct. 2014

10. **Pooria M. Yaghini**, Ashkan Eghbal, Hossein Pedram, and H. R. Zarandi, "Investigation of Transient Fault Effects in Synchronous and Asynchronous Network on Chip Router,". *Journal of Systems Architecture*, Volume 57, Issue 1, Pages 61-68, January 2011.
11. Ashkan Eghbal, **Pooria M. Yaghini**, Hossein Pedram, and H. R. Zarandi, "Designing a Fault-tolerant NoC Router Architecture,". *International Journal of Electronics*, Volume 97, Issue 10, Pages 1181-1192, October 2010.
12. S. A. Asghari, Hossein Pedram, M. Khademi, and **Pooria M. Yaghini**, "Designing and implementation of a network on chip router based on handshaking communication mechanism," *World Applied Sciences Journal*, Volume 6, Issue 1, Pages 88-93, 2009.
13. H. Aliee, H. R. Zarandi, and **Pooria M. Yaghini**, "K2Router: A Low-Power and High-Performance Router Design for Networks-On-Chip," *Journal of Computer Science and Engineering*, Volume 7 Issue 2, Pages 8-23, 2011.
14. **Pooria M. Yaghini**, Ashkan Eghbal, Hossein Pedram, and Hamid R. Zarandi, "Investigation of Transient Fault Effects in an Asynchronous NoC Router," in *Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on*, pp. 540545, IEEE, 2010.
15. Ashkan Eghbal, **Pooria M. Yaghini**, Hossein Pedram, and Hamid R. Zarandi, "Fault Injection-based Evaluation of a Synchronous NoC Router," in *On-Line Testing Symposium, 2009. IOLTS 2009. 15th IEEE International*, pp. 212214, IEEE, 2009.
16. Ashkan Eghbal, Hamid R. Zarandi, and **Pooria M. Yaghini**, Fault-Tolerance Assessment of PIC Microcontroller based on Fault Injection, in *Test Workshop, 2009. LATW09. 10th Latin American*, pp. 16, IEEE, 2009.
17. **Pooria M. Yaghini**, Ashkan Eghbal, Amir Asghari, and Hossein Pedram, Power Comparison of an Asynchronous and Synchronous Network on Chip Router, in *Computer Conference, 2009. CSICC 2009. 14th International CSI*, pp. 242246, IEEE, 2009.
18. **Pooria M. Yaghini**, Hamid R. Zarandi, Ashkan Eghbal, Akbar Jafarzadeh, and S. Eskandari, An Investigation of Fault-Tolerance Behavior of 32-bit DLX Processor, in *Dependability, 2009. DEPEND'09. Second International Conference on*, pp. 9398, IEEE, 2009.

# ABSTRACT OF THE DISSERTATION

Resilient 3D Network-on-Chip Design and Analysis

By

Pooria Mohammadi Yaghini

Doctor of Philosophy in Electrical and Computer Engineering

University of California, Irvine, 2016

Professor Nader Bagherzadeh, Chair

Like every other major changes in computer architecture, exascale computing, targeted for 2020, requires dramatic and unanticipated shifts in different perspectives. The biggest challenge facing this trend is to design an exascale system with a hundredfold optimization on the estimated power cost of above \$2.5B per year for a system designed with current technology. It has been reported that a large portion of total power is consumed for communication through interconnection network. Communication between the computational components of System-on-Chip (SoC) designs can account for more than 25 percent of the energy dissipation of the whole system. NoC is recognized by many researchers as the best communication infrastructure for many-core systems. To lower communication power, researchers have proposed the idea of designing thinned and stacked 3D ICs. 3D ICs, fabricated using Through-Silicon Via (TSV), offer higher bandwidths, smaller form factors, shorter wire lengths, lower power, and better performance than traditional 2D ICs. The combination of 3D structures and NoC is the most promising approach for obtaining the projected performance and power requirements for exascale systems. Besides the extremely constrained power



budget, achieving an acceptable level of resiliency for 1,000,000 cores in an exascale system is a crucial challenge. Communication reliability, due to the huge amount of data movement in these systems, plays a key role.

In this dissertation, the focus is to identify, characterize, and mitigate the reliability threats of TSV-based 3D communication structures, specifically threats introduced by TSV-to-TSV coupling fault.

In the first step which is the identification of the reliability threats, the potential physical faults of a baseline TSV-based 3D NoC architecture by targeting Two-dimensional (2D) NoC components and their inter-die connections is classified. Subsequently, TSV issues, thermal concerns, and Single Event Effect (SEE) are investigated and categorized, in order to propose evaluation metrics for inspecting the resiliency of 3D NoC designs.

Then, in the second step, having overviewed the common TSV issues, a framework is proposed for quantifying the 3D NoC reliability using formal methods. TSV issues are modeled as a time-invariant failure probability and a reliability criterion for TSV-based NoC is defined. The relationship between NoC reliability and TSV failure is quantified. For the first time, the reliability criterion is reduced to a tractable closed-form expression that requires a single Monte Carlo simulation.

In the third step, a system-level TSV coupling fault model is proposed, which models the capacitive coupling effect, considering thermal impact, at circuit-level accuracy. This model can be plugged into any system-level and RTL-level TSV-based 3D-IC data-oriented simulator. Having analyzed and recorded the TSV coupling effect at circuit-level, these effects are applied to the Through-Silicon Vias (TSVs) dynamically

in system-level simulations at runtime through precise monitoring and calculation. The proposed fault model is potentially useful for evaluating the reliability of 3D many-core applications in which TSV coupling may lead to failure.

After setting up the TSV coupling fault modeling framework, multiple coding approaches are proposed to prevent coupling fault occurrence on TSV links. In these approaches, the coupling fault effect is addressed by diagnosing the hazardous current flow direction patterns of the TSV bus, and encoding the data bits to avoid those patterns at run-time. Different coding schemes are devised to address both types of TSV coupling, inductive and capacitive. These approaches are devised to be low overhead, fast, and highly efficient. Empirical simulations are performed with both random and realistic benchmarks, including PARSEC, to demonstrate the efficacy of the devised approaches. All these approaches are also implemented at hardware-level, to have a realistic estimate of the imposed overheads at logic-level. Experimental results show that these approaches improve the communication reliability over TSV links significantly, with no extra TSV and negligible information redundancy or hardware logic overhead.

Overall, this work provides a rich set of TSV coupling-avoidance techniques, besides an accurate and fast TSV coupling fault modeling simulation framework, for efficient and effective design of reliable 3D communication architectures. It helps DFT designers to more easily design robust TSV links.

# Chapter 1

## Introduction

Every major shift in computer architecture has led to changes, usually dramatic and often unanticipated, and the move to exascale computing will be no exception. At the hardware level, feature size in silicon will almost certainly continue to decrease at the Moore's law pace by the end of this decade. To continue efficiency both in high performance computing systems and consumer electronics, a fundamental change is required for the design of next generation computer systems. Exascale computer systems are needed for the growing number of problems where experiments are impossible, dangerous, or extremely expensive. These machines, along with parallel computing, will enable the analysis, modeling and processing of enormous amount of data, leading to advances in various areas of science and technology.

Exascale computing is challenging due to the strict constraints in power requirements, requiring new communication infrastructure and software approaches to exploit parallelism and scalability. Based on the current technology, scaling today's systems

to an exaflop level would consume more than a gigawatt of power [79]. Reducing the power requirement by a factor of at least 100 is a challenge for both future hardware and software technologies.

Today's top supercomputer systems include approximately 0.5 to 3 million cores [1]. By 2020, due to design and power constraints, the clock frequency is unlikely to change, which means that a High Performance Computing (HPC) system will have approximately one billion cores [76][21][77]. An immediate consequence is that the power consumption and hence temperature will increase while timely power and thermal management become much more difficult. Mathematical models, numerical methods, and software implementations will all need new conceptual and programming paradigms to make effective use of unprecedented levels of concurrency.

The applications of exascale computing include, but not limited to:

- Adaptation to regional climate changes such as sea level rise, drought, and flooding;
- Weather forecasting;
- Reverse engineering of the human brain;
- Finite-Difference Time-Domain (FDTD) numerical computation;
- Methods of Moment (MoM) numerical computation;
- Design, control and manufacturing of advanced materials.
- Big data processing.

The increase in the number of components for exascale computing is faster than reliability improvement pace of those components. Based on observations of existing HPC systems, it is projected that the Mean-Time Between Failure (MTBF) of an

exascale system to be in minutes or seconds, caused by various kinds of faults. The cause for these faults are varied including the effects from terrestrial neutrons, naturally occurring alpha particles, electromagnetic interference, temperature, and voltage fluctuations. These faults are mostly transient, but permanent ones are also seen. Obviously, there is a need for any low-overhead methods to increase MTBF and that can detect, and if possible correct, errors at the hardware level. In order to embark on this, fault-tolerant approaches require a knowledge of the actual errors occurred on HPC systems, the rate of these errors, and, ideally, the most common source of faults. This knowledge is currently unknown for the new technologies; hence there is an immediate need to develop models that are required to reason about fault avoidance, detection and correction. Also novel fault injection tools are required to verify the efficacy of proposed fault tolerance techniques, along with accurate and fast models of faults. Put simply, correctness assurance and resiliency will be crucial to make exascale machines worth their high usage cost.

Wiring delay is one of the critical issues of the sub-micron level progress in semiconductor processing technology. Increasing the capacitance of gates is a result of narrow channel width and crosstalk phenomenon; the wiring delay is growing exponentially while the speed of basic elements such as gate delay becomes much faster. According to [38], in 2018 the nominal logic gate delay of a high performance NAND gate is estimated to be less than 3 ps while the interconnect RC <sup>1</sup> delay for a 1 mm copper global wire is to be more than 3500 ps. In other words, the interconnect delay is estimated to be 1000 times greater than the gate delay, as shown in Figure 1.1. Therefore, if this trend continues, the wiring delay will be one of the most critical issues for future designs.

---

<sup>1</sup>Resistance-Capacitance

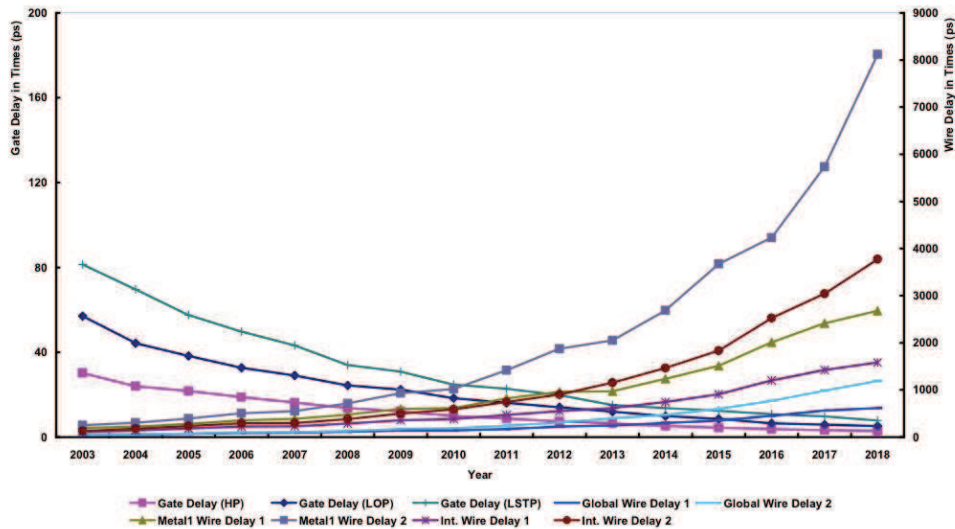


Figure 1.1: Wire delay versus gate delay [38]

3D architecture is a promising approach for future IC design which addresses some of the wire delay constraints expected from the next generation process technologies. At the moment, a major paradigm shift from 2D technology to 3D technology, is occurring in the microelectronic industry [49], [122], [100], [68] and [40]. 3D or vertical integration is an exciting path to boost the performance and extend the capabilities of modern integrated circuits. These capabilities are inherent to 3D ICs. The former enhancement is due to the considerably shorter interconnecting wires in the vertical direction. It is also worth noting that vertical integration is particularly compatible with the integrated circuit design process that has been developed over the past several decades. These distinctive characteristics make 3D integration highly attractive as compared to other radical technological solutions that have been proposed to resolve the increasingly difficult issue of on-chip interconnect. One of the most promising technologies for 3D IC integration is the notion of TSVs [107], pillars manufactured across thinned silicon substrates to establish inter-die connectivity after die bonding.

Salient TSVs features include fine pitches, high densities, and high compatibility with the standard CMOS process. TSV-based 3D ICs are interconnected with high-density short and thin TSVs, supporting low-level integration, superior to existing solutions. There exists different process technologies to interconnect the 3D IC, like photonics, or inductive; in this thesis we only focus on TSV-based 3D ICs.

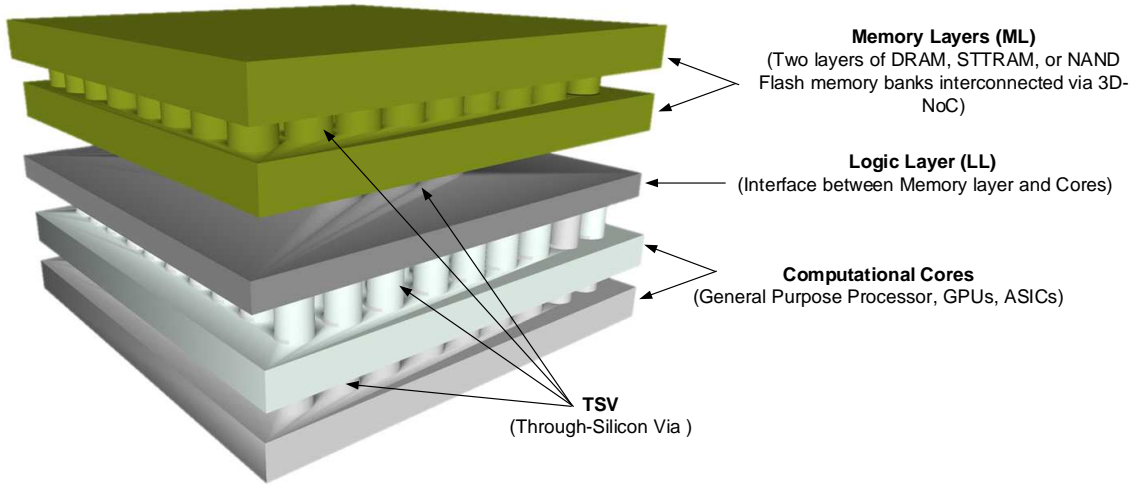


Figure 1.2: Processing Memory Elements (PME), as an example of 3D IC. This architecture depicts a 3D IC with memory, CPU, GPU, and interconnection network stacked on top of each other.

It has been reported that a large portion of total power is consumed for communication in interconnection network. Communication between computational components on SoC designs can account for more than 25 percent of the energy dissipation of the whole system in exascale systems [14]. Therefore, it is paramount that the communication hardware of the high performance system is designed in an energy efficient way.

NoC (Figure 1.3) is the dominant infrastructure for communication in highly massive manycore systems. Over the years, microprocessors continue to improve, following

Moore's law. System integration has reached a stage where a complete system can be integrated onto a single chip. As SoC design expands to encompass ever-increasing cores to meet the high performance needs, it has become clear that traditional bus-based systems do not provide scalability and efficiency in interconnecting large number of cores on a chip.

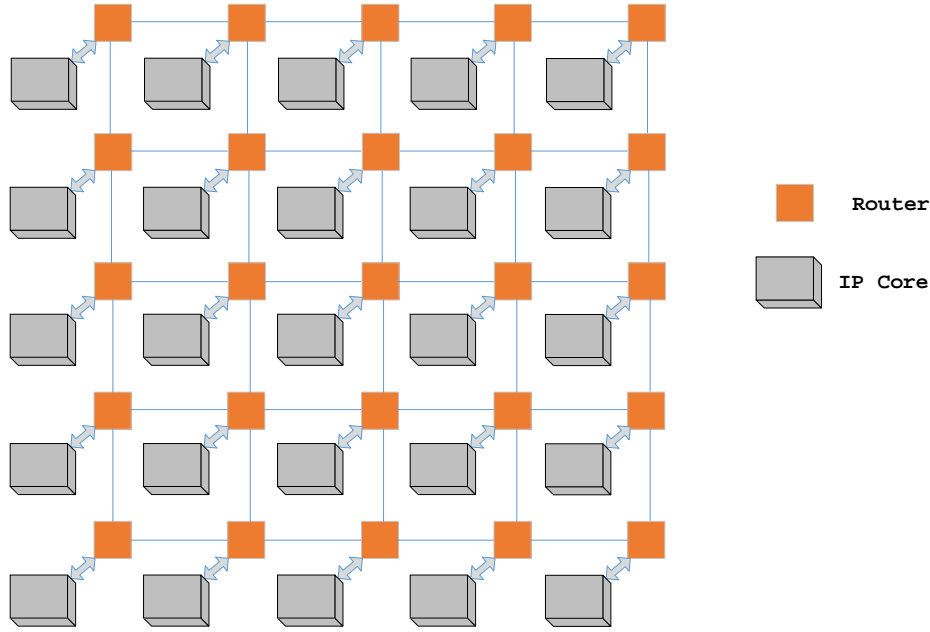


Figure 1.3: NoC architecture;  $5 \times 5$  mesh topology

NoC is an infrastructure composed of computational processors and memories, or Intellectual Properties (IP)s in general, that are connected with network of routers. Resources communicate with each other using packets that are routed through the network as the conventional Internet network does. One of the advantages of NoC over traditional bus-based systems is that it decouples communication from computation and reduces the design complexity. For example, routers, as illustrated in Figure 1.3, in a NoC are responsible only for packet switching and cores handle computations. This leads to additional advantages for a NoC which are reusability, scalability, and



flexibility. They also have the ability for fast prototyping of an SoC design as IP cores can be inserted in or replaced with other ones easily just by adding a wrapper for supporting an NoC.

Combining 3D technology and NoC would lead to less power consumption and higher performance due to the shorter links used in vertical direction. The combination of 3D integration and NoC technologies provide a new horizon for on-chip interconnect design. The major advantage of 3D-NoC is the considerable reduction in the length and number of global interconnects, resulting in performance increase and power consumption decrease.

On the other hand, as CMOS technology continues to shrink, there is an increasing need for studying how NoC architectures can tolerate faults and the underlying process variations [2]. Shrinking transistor sizes, smaller interconnect features, 3D packaging issues, and higher frequencies of CMOS circuits, lead to higher soft-error rates and more thermal and timing violations.

As the fundamental concepts of this dissertation, the required basics about resilience and NoC is explained in the rest of this thesis.

## 1.1 Basics of Resilience

Reliability is a conditional probability that a system will perform correctly throughout the interval  $[t_0, t]$ , given that it was performing correctly at time  $t_0$ , directly related to the continuity of service. Selection of the right mechanisms to integrate in a circuit requires definition of fault types that could occur and a detailed knowledge of their

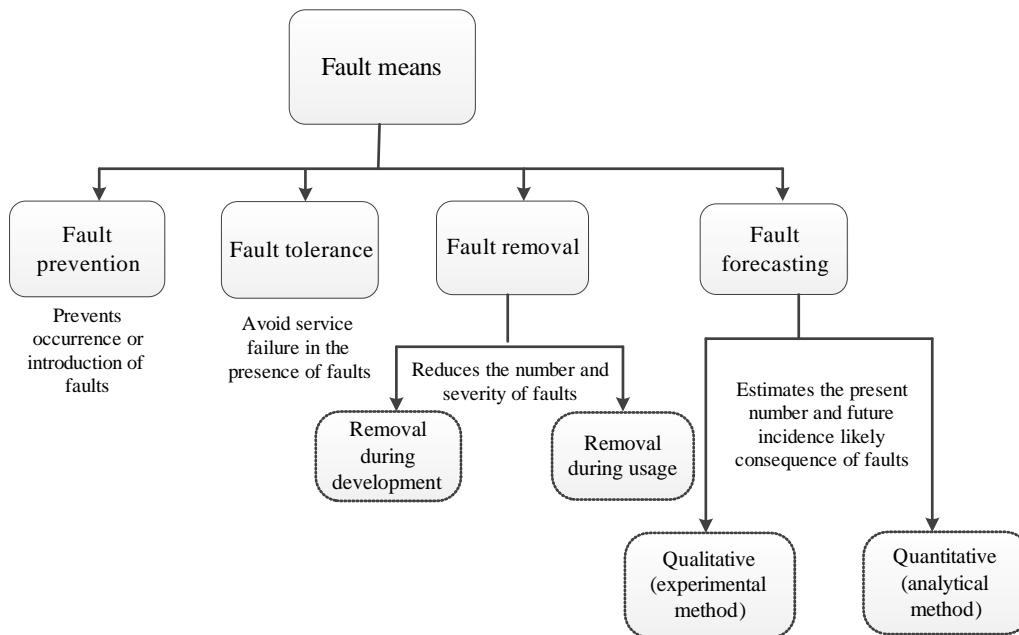


Figure 1.4: Fault means to reach dependability

potential impact on the circuit behavior. Causes and consequences of deviations from the expected function of a system are called threats to dependability, which includes various faults, errors, and overall failure of the system.

- **Fault** is a physical defect, imperfection, or flaw that occurs within some hardware or software component.
- **Error** is a deviation from accuracy or correctness and is the manifestation of a fault.
- **Failure** is the non-performance of some actions that are expected.

An incorrect change in a state machine because of a fault may result in an error. Multiple errors can be originated from a single fault and propagate throughout the system, while a fault remains localized in the affected circuit. Any of these errors can

make the system fail [6]. Accurate definition of various fault types in the 3D-NoC environment is necessary to model and simulate the possible physical faults. Figure 1.4 shows different methods of reaching dependability. Fault prevention deals with preventing faults from impacting correct system operation. This can be accomplished by use of development methodologies and good implementation techniques, but they cannot completely eliminate the risk of faults. Fault removal can be sub-divided into two sub-categories: removal during development and removal during use. Removal during development requires verification so that faults can be detected and removed before a system is put into production. Once a system has been put into production, a mechanism is needed to record failures and remove them using a maintenance cycle to support removal during use method. Fault forecasting predicts likely faults so that they can be removed or their effects can be circumvented. Fault tolerance established mechanisms that will allow a system to still deliver the required service in the presence of faults, although that service may be at a degraded level.

In most cases, the expected dependability is not achieved by using any of the fault means separately. Different fault-tolerant techniques have been introduced by researchers [104, 20, 36], but it would be more effective if the target device is fully examined to minimize the overhead of any proposed fault-tolerant method.

There is a need to consider fault forecasting methods in order to address the vulnerable components of a system; expected dependability is met while redundancies cost is kept low. In practice, fault removal and fault forecasting methods are followed by each other. In other words, after rejecting a system by the fault forecasting, several fault removal tests are applied. These new tests provide actions that can help the designer to improve the system. This loop is repeated until reaching the desirable

design. Quantitative (analytical) and qualitative (experimental) methods are well-known fault forecasting methods. Quantitative methods commonly have impractical assumptions. Furthermore, there is a need to determine suitable input parameters for formal analysis techniques. Qualitative techniques are more popular, but require more time for calculations. One major drawback of this method is the accuracy of the simulation which is confirmed by running different benchmarks. In literature, quantitative techniques have been known as a complementary method. Fault injection is one of the popular qualitative techniques [120, 116, 31, 33]. The failure scenario generation is quite complicated to be managed, so a fault injector tool is needed to validate devised fault tolerance behavior and potential failure points of a system [10].

In this dissertation, fault prevention and fault forecasting methods are being utilized to enhance reliability of 3D-NoC systems.

## 1.2 Basics of NoC

NoC is the dominant infrastructure for communication in highly massive manycore systems. The baseline NoC router, utilized in this dissertation, is designed with five bidirectional ports, to support mesh and torus topologies. It is composed of three main components including an input buffer, a management and routing unit, and a crossbar switch [32, 117, 5, 113, 4], as shown in Figure 1.5. The shaded component depicts the management and routing unit of the router.

The input buffer is responsible for storing incoming packets while there is free space. By using a circular buffer scheme, it is possible to optimize the buffering oper-

ation. Buffer management subcomponent is composed of two-state FSM for receiving, storing and transferring data packets, as illustrated in Figure 1.5. To comply with the wormhole switching technique, the buffer size is chosen to be less than the packet size.

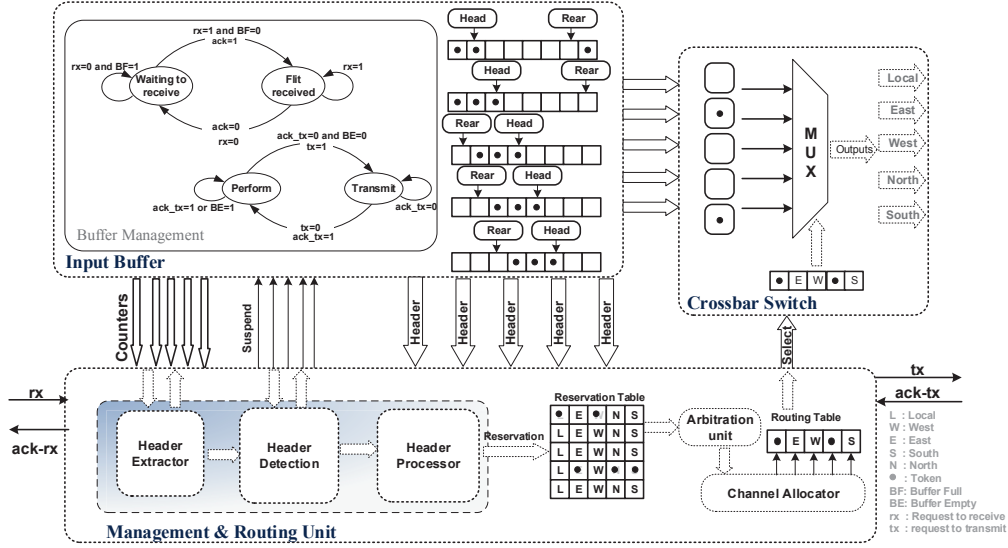


Figure 1.5: NoC router architecture for distributed routing

The management and routing unit component is the central unit which includes header extractor and header processor for handling the routing function, arbitration and updating the routing table. Once the trailer flit is recognized by the routing component, the corresponding output port is released. The grant and activation signals are disabled as the trailer flit of a packet is transmitted to its desired output port. Each of the input channels can reserve one of the output ports when the routing process is accomplished. An arbitration unit locks the dedicated output channel until the end of packet transmission. It is also responsible for assigning output channels evenly among input ports by implementing a round robin algorithm. The routing unit grants the requested input port and triggers the selected output multiplexer to establish a

connection, once the routing process is accomplished successfully. Such grant and activation signals are disabled as the trailer flit of a packet is transmitted.

The third component of an NoC router is the crossbar switch. Control signals of the routing unit select one of the output ports for an incoming header flit. By implementing the wormhole switching, all of the remaining flits of a packet follow the header flit. If a header is blocked, then the following flits are blocked, as well. Once a packet transmission is completed, the switch is unlocked to serve other input channels.

### 1.3 Dissertation Contributions

Like every other major changes in computer architecture, exascale computing, targeted for 2020, requires dramatic and unanticipated shifts in different perspectives. The biggest challenge facing this trend is to design an exascale system with a hundredfold optimization on the estimated power cost of above \$2.5B per year for a system designed with current technology. It has been reported that a large portion of the total power is consumed for communication through the interconnection network. Communication between the computational components of SoC designs can account for more than 25 percent of the energy dissipation of the whole system. In the same context, NoC is recognized by many researchers as the best communication infrastructure for manycore systems. To lower communication power, researchers have proposed the idea of designing thinned and stacked 3D ICs. 3D ICs, fabricated using TSV, offer higher bandwidths, smaller form factors, shorter wire lengths, lower power, and better performance than traditional 2D ICs. The combination of 3D structures and NoC is the most promising approach for obtaining the projected performance and power

requirements for exascale systems. Besides the extremely constrained power budget, achieving an acceptable level of resiliency for millions of cores in an exascale system is a crucial challenge. Communication reliability, due to the huge amount of data movement in these systems, plays a key role. This dissertation aims at identifying, characterizing, analyzing and modeling the reliability threats in emerging TSV-based 3D communication structures caused because of vertical stacking.

This dissertation's contributions can be summarized as follows:

- To highlight potential sources of physical faults in 3D-NoC and present their corresponding logic-level fault models.
- To address the impacts of all potential physical faults on 3D-NoC components. It provides reliability metrics for 3D-NoC environment.
- To analyze the reliability of 3D-NoC using quantitative (analytical) method.
- To devise a system-level TSV-to-TSV coupling fault model that models the capacitive coupling effect, considering thermal impact, with circuit-level accuracy.
- To design and develop coding approaches to mitigate TSV-to-TSV inductive and capacitive coupling faults.
- To design an asynchronous architecture to prevent TSV-to-TSV capacitive coupling faults.

## 1.4 Dissertation Organization

This dissertation is organized as follows:

After the introduction to the research background of this work in Chapter 1, a comprehensive study on the reliability issues of 3D integration, and their effects on the 3D-NoC architectures is presented in Chapter 2. This chapter classifies the potential physical faults of a baseline TSV-based 3D-NoC architecture by targeting 2D NoC components and their inter-die connections [27, 30]. In this chapter, TSV issues, thermal concerns, and SEE are investigated and categorized, in order to propose evaluation metrics for inspecting the resiliency of 3D-NoC designs. Chapter 2 lays the foundation for the research in this dissertation.

Chapter 3 aims to model TSV characteristics as a time-invariant failure probability [53]. In this chapter, a reliability criterion for TSV-based NoC is defined as the probability of having at least one faulty TSV in a given time slot. Consequently, the relationship between NoC reliability and TSV failure is quantified, and the resulting equation is reduced to a tractable form with manageable computational complexity. The final equation relating the NoC reliability criterion and TSV failure includes a non-analytical probabilistic term which can be efficiently approximated by the Monte Carlo simulation for different architectures. Importantly, the simulation only depends on network geometry and routing algorithm, and the effect of injection rate and TSV failure is decoupled from the simulation. Therefore, the result of simulation can be used to calculate the reliability criterion for a wide range of injection rate and TSV failure rate values.



Chapter 4 proposes a system-level TSV-to-TSV coupling fault model that models the capacitive coupling effect, considering thermal impact, with circuit-level accuracy. This model can be plugged into any system-level TSV-based 3D-NoC simulator [118]. It is also capable of identifying faulty TSV bundles and evaluating the efficiency of alternative resilient TSV-based 3D-NoC designs at the system-level. The presented model can be utilized for application-specific designs by addressing the susceptible to failure TSVs. With these results a designer is able to employ fault-tolerant methods only where they are required. For general purpose architectures, the presented fault model is able to figure out the effect of physical parameters of TSVs on timing requirement of the circuits. This model can be used to find the suitable physical parameters for a TSV to have reliable TSV links.

In Chapter 5, first the inductance parasitics in contemporary TSVs is characterized, and then a classification for inductive coupling cases is analyzed and presented [28]. Next, a coding algorithm is devised to mitigate the TSV-to-TSV inductive coupling. The coding method controls the current flow direction in TSVs by adjusting the data bit streams at runtime to minimize the inductive coupling effects. After performing formal analysis on the efficiency scalability of devised algorithm, an enhanced approach supporting larger bus sizes is proposed. The experimental results show that the proposed coding algorithm yields significant improvements, while its hardware-implemented encoder results in tangible latency, power consumption, and area.

In Chapter 6, two coding methods, baseline and enhanced, is proposed in order to minimize the TSV-to-TSV capacitive coupling effect [29]. Baseline algorithm is proposed for small mesh of TSVs which are considered in 3D-NoC applications, while

enhanced method is suggested for large mesh of TSVs which are more applied in 3D memory applications. The enhanced method guarantees that the encoding process eliminates undesirable parasitic capacitance values by recognizing all susceptible configurations. According to experimental results, the baseline method's mitigation rate is more than 90% for TSV meshes smaller than  $10 \times 10$ . The enhanced algorithm mitigates the TSV-to-TSV capacitive coupling more than 70% for  $8 \times 32$  mesh of TSVs.

In Chapter 7, a novel architectural approach (hardware redundancy) is devised to effectively reduce the effect of TSV-to-TSV capacitive coupling [119]. This approach also helps in reducing the utilized TSVs, which would be cost-saver. This approach, in abstract, is using dual-rail coding combined with multiple lane Quasi Delay Insensitive (QDI) asynchronous Serializer-Deserializer (SerDes) to exploit the high-performance feature of TSV and compensate the inevitable extra bits imposed by dual-rail coding.

# Part I

## Fundamentals, Analysis and Modeling

## Chapter 2

# Three-dimensional Networks-on-Chip Reliability Concerns

Reliability is one of the most challenging problems in the context of 3D-NoC systems. Reliability analysis is prominent for early stages of the manufacturing process in order to prevent costly redesigns of a target system. This chapter classifies the potential physical faults of a baseline TSV-based 3D-NoC architecture by targeting Two-dimensional Networks-on-Chip (2D-NoC) components and their inter-die connections. TSV issues, thermal concerns, and SEE are investigated and categorized in this chapter, in order to propose evaluation metrics for inspecting the resiliency of 3D-NoC designs. In this chapter, reliability analysis for major sources of faults are reported based on their MTBF. TSV failure probability induced by inductive and capacitive coupling is also discussed in this chapter.

## 2.1 Motivation

Technology scaling, improving transistor performance with higher frequency, designing novel architectures, and reducing energy consumed per logic operation have become essential for improving the computational performance by orders of magnitude. Furthermore, energy efficiency is of great importance for both future supercomputers and embedded systems [78]. Reliability is another significant challenge of single chip designers as petascale computational performance comes to fruition, by targeting exascale systems for the next decade [121]. Increasing power consumption, power variation, and power density have negative impacts on reliability of on-chip designs. Rapid changes in power consumption, uncovers on-chip voltage fluctuations and consequently leads to transient errors. High temperature also increases leakage power consumption, resulting in a self-reinforcing cyclic dependency between power and temperature [69, p. 25]. On the other hand, higher chip temperatures may be derived from high temporal and spatial power densities. Power consumption and consequently temperature have a direct relationship with the reliability of systems [16].

With technology and design scaling slowing down, the processor industry is rapidly moving from a single core with high-frequency designs to manycore chips. 3D integration instead of 2D integration is another trend to keep the traditionally expected performance improvements. These computational processors and memories, or IPs in general, need robust, high performance and low power interconnections. Over the years, system integration has reached a stage where a complete system can be integrated onto a single chip. As SoC design expands to encompass ever-increasing cores to meet the high performance needs, it has become clear that traditional bus-based

systems do not provide scalability and efficiency in interconnecting large number of cores on a chip. Therefore, NoC has been proposed as a scalable and efficient interconnection for future systems [91]. The combination of 3D integration and NoC technologies provides a new horizon for on-chip interconnect design. The major advantage of 3D-NoCs is the considerable reduction in the length and number of global interconnects, resulting in higher performance and lower power consumption [70]. Shrinking transistor sizes, smaller interconnect features, 3D packaging issues, and higher frequencies of CMOS circuits, lead to higher error rates and more thermal and timing violations [32, 70]. Reliable NoC architectures have been introduced in various articles. Many fault-tolerant routing algorithms have been proposed for both 2D [25] and 3D-NoC [82]. The idea of bypassing faulty data paths within failed routers has been suggested as a lightweight fault-tolerant method [56]. A novel bidirectional fault-tolerant NoC architecture capable of mitigating both static and dynamic channel failures is also proposed [104]. However, they all improve the reliability of NoC design, regardless of imposing hardware, software, or time redundancy. Experimental and analytical techniques are popular methods in order to explore the reliability of any system. Categorizing the effects of potential faults on the performance of a system is needed for both. A dependability analyzer of a system should be completely familiar with the design to explore the sensitivity of each of the components against potential source of faults [32].

In this chapter, potential sources of physical faults in 3D-NoC are highlighted and their corresponding logic-level fault models are presented, which is reported in Section 2.3. Then, the impacts of all potential physical faults on 3D-NoC components are addressed. It provides reliability metrics for 3D-NoC environment, as discussed in

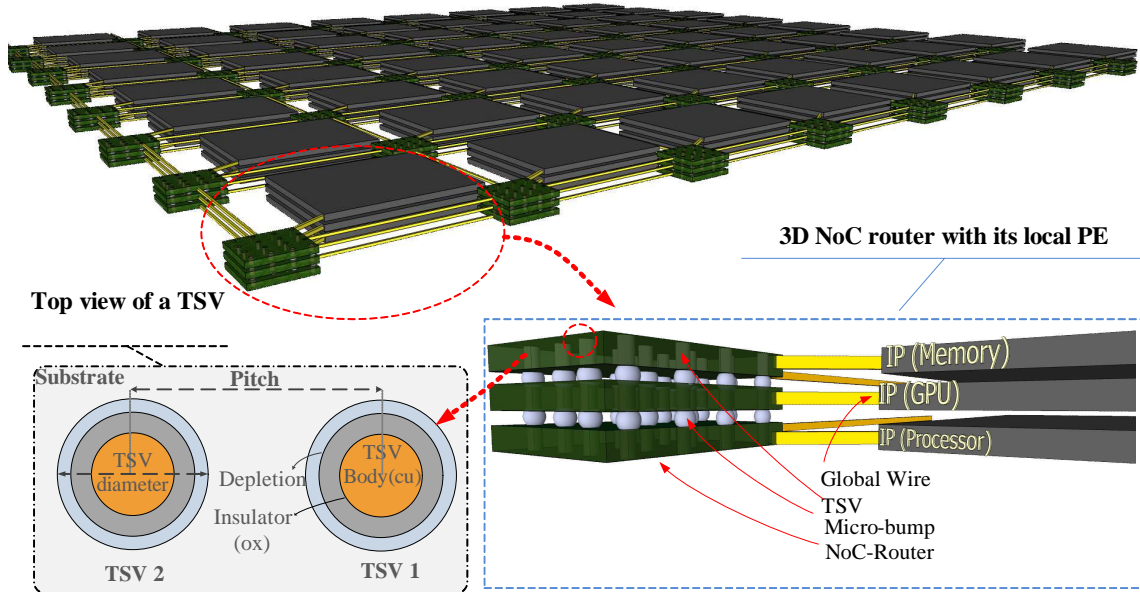


Figure 2.1: 3D-NoC structure

Section 2.4. Also, a reliability study for the physical faults in 3D-NoC is elaborately detailed in Section 2.5.

## 2.2 3D-NoC Architecture

NoC is the only scalable communication structure for manycore systems with hundreds of cores known to date. NoC offers higher flexibility and modularity, supporting simpler interconnect models with higher bandwidth as compared traditional SoC approaches. A baseline NoC router design is composed of three main components including an input buffer, a routing unit, and a crossbar switch, which are discussed in detail in Section 1.2.

As stated earlier in Section 1, the introduction of vertical integration enables

higher density manycore architectures to take place and helps in boosting up the power-performance characteristics to extend capabilities of modern integrated circuits [26, 89, 22, 81]. These capabilities are inherent to 3D ICs, resulting in considerably shorter interconnecting wires in the vertical direction. 3D integration supports new opportunities by providing feasible and cost effective approaches for integrating heterogeneous cores to realize future computer systems. It supports heterogeneous stacking because different types of components can be fabricated separately, and silicon layers can be implemented with different technologies. One of the most promising technologies for 3D IC integration is the notion of TSVs [105], pillars manufactured across thinned silicon substrates to establish inter-die connectivity after die bonding. Principal TSV features include: fine pitches, high densities, and high compatibility with the standard CMOS process. Multiple layers of 2D planar designs are stacked together and are vertically interconnected by high-density short and thin TSVs in 3D integration technologies as shown in Figure 2.1.

Micro-bumps are the interface between TSVs and 2D layers. The maximum height of TSV is about  $200\mu m$  or less which is the same as thickness of the Silicon (Si) chip. The diameter of vias in TSV is now  $20\mu m$  and may reach  $5\mu m$  in future [105]. A copper TSV in standard Si-bulk technology normally will have via diameter of  $2\mu m$ - $8\mu m$  by 2018,  $5\mu m$  by  $5\mu m$  contact pads,  $4\mu m$ - $16\mu m$  via pitch,  $0.5\mu m$  oxide thickness, ( $t_{ox}$ ), and  $20\mu m$ - $50\mu m$  layer thickness including substrate and metallization [42]. It is important to note that the TSV fabrication process is independent of the CMOS fabrication technology and TSVs do not scale down with the same pace. TSV diameters and pitches are two to three orders bigger than transistor gate lengths.



## 2.3 Reliability Concerns in 3D-NoC

The need for reliability assessment in order to figure out the underlying process variation of NoC architecture has become more critical as CMOS technology continues to shrink [64]. This is because the anticipated fabrication geometry in 2018 scales down to  $8nm$  with projected  $0.6v$  supply voltage [42]. Such a small supply voltage is close to the operating point with minimal energy consumption. In the  $8nm$  process, higher rate of soft errors impact buffers and control logic of NoC routers dramatically, leading to error(s) and consequently a chip failure. The low voltage supply enforces a very narrow noise margin which makes the architecture vulnerable and sensitive to faults. With rising power density and non-ideal threshold and supply voltage scaling, transient soft errors become increasingly common during a chip's lifetime.

On the other hand, a major paradigm shift from 2D technology to 3D, is currently being pursued in industry [50, 41]. Vertical integration is particularly compatible with the integrated circuit design process which has been developed over the past several decades. These distinctive characteristics make 3D integration highly attractive as compared to other radical technological solutions that have been proposed to resolve the increasingly difficult issues of the on-chip interconnections. Many challenging issues are expected due to the distributions of the current density, temperature and stress in a 3D structure from the device reliability perspective. Mass production of 3D ICs for consumer electronic products are not reliable enough without a systematic and thorough reliability assessment. Evaluating the reliability of 3D interconnection architecture at the logic-level enforces the necessity for a fault forecasting method. Quantitative (analytical) and qualitative (experimental) techniques are well-known

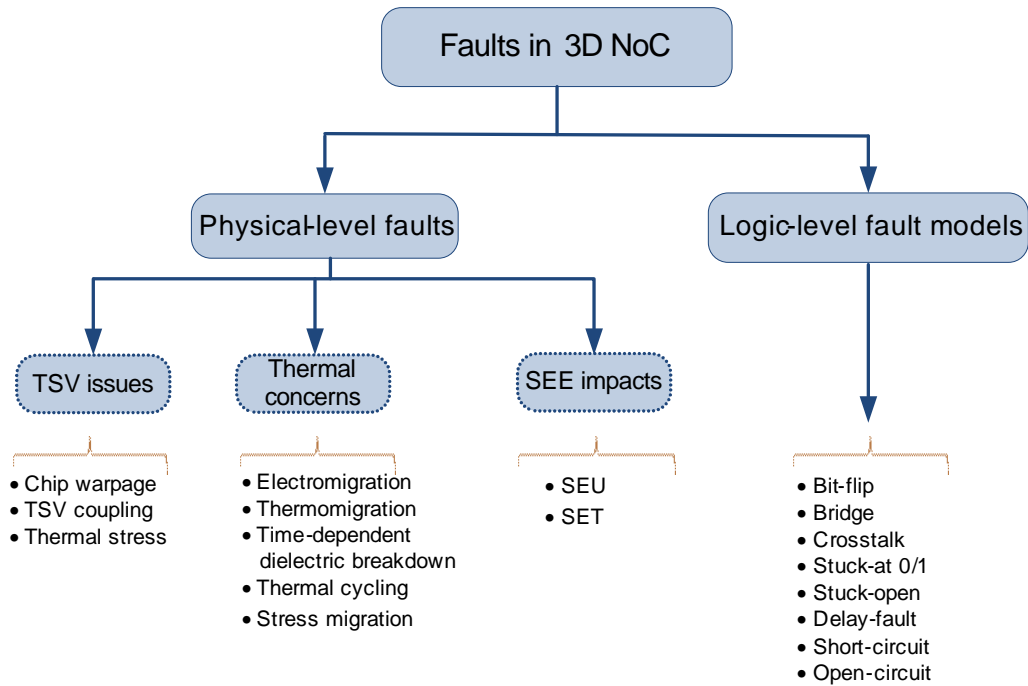


Figure 2.2: Potential physical-level faults in 3D-NoC and their corresponding logic-level models

fault forecasting methods. Both of them need a comprehensive study on the sources of faults and their effects.

Figure 2.2 summarizes the potential physical faults, affecting the performance of 3D-NoC design and divides them into subcategories. However, physical-level simulation of the target model is time-consuming while it is not accurate enough at system-level. Logic-level fault models are also listed in Figure 2.2 to represent corresponding physical faults. Such a categorization provides accurate and efficient simulation technique for fault injection and signal observation in experimental methods. It is also useful for providing more realistic formal equations for analytical techniques.

Following subsections provide a description of the potential physical faults and their effects on a 3D-NoC design and introduce corresponding logic-level fault models

for each of physical faults.

### 2.3.1 Physical-level Potential Faults

At physical-level, a failure mechanism is the mechanical or chemical action that actually causes the manufactured circuit to act different than desired. The resulting damage is called failure mode. TSV issues, thermal concerns, and SEE impacts are the main physical sources of faults in future 3D-NoC designs [105, 96, 87]; it is unlikely that existing technology becomes ubiquitous in the near future until there are solutions for these issues [42]. These physical fault sources influence behavior of 3D-NoC during packet transmission. Thermal concern has effects on both 2D planar and vertical links, while SEE impacts only target transistor of 2D planar designs. Each of these physical faults is explained individually in the following subsections.

- **TSV issues:** 3D-NoCs are expected to offer various benefits such as higher bandwidth, smaller form factor, shorter wire length, lower power, and better performance than traditional 2D-NoCs. However, they are sensitive to the introduced sources of physical faults in Figure 2.2. The impact of sub-micron TSVs on future 3D-NoCs are still unknown [54]. A reliability analysis is needed to investigate their unexpected effects on the 3D-NoC design. Chip warpage, TSV coupling, and thermal stress are known as the main causes of TSV failure [105, 42]. Some of the thermal issues other than thermal stress may also affect the functionality of TSVs which are addressed in thermal concerns subsection.

### **Chip warpage**

When TSVs are used for vertical interconnection, it may lead to chip damage if TSVs are arranged in a non-uniform manner. The other reason is that the thermal expansion of Si and Cu are different, resulting in chip compression stress. Typically, TSVs are placed on the peripheral or the center of a chip. TSV-related defects might occur in the fabrication process of the TSV placement, in the bonding of TSVs to the next layer like wafer warpage [105]. The wafer warpage is considered a fabrication defect as a result of the annealing process.

### **TSV coupling**

According to recent results, the occupied die area by TSVs is quite significant, which in turn compromises the wire length benefit of 3D ICs. In addition, TSV capacitance increases the latency of 3D signal paths. Although buffer insertion reduces the delay overhead it also demands additional silicon area and power. The degree of negative effects of TSVs depends on fabrication technology and physical design parameters. Furthermore, small TSVs can have large capacitance values depending on the liner thickness and doping concentration of the substrate. In this case, small TSVs may not cause area overhead, but they cause serious delay overhead. As technology size scales down to  $8nm$ , TSV coupling becomes critical due to tighter timing requirements. In other words, TSV coupling may result in delay or even mutual coupling between adjacent TSVs [42, 62]. The term TSV coupling refers to capacitive and inductive couplings among neighbor TSVs. Electric field results in capacitance coupling and magnetic field is a source of inductive coupling.

## Thermal stress

The mismatch between the Coefficient of Thermal Expansion (CTE) of a TSV fill material and the silicon induces a residual thermal stress in the region surrounding the TSV [72, 71]. The thermal stresses can drive interfacial delamination between the TSV and the Si matrix, damaging for the on-chip wiring structures [95, 90]. It can affect the carrier mobility due to the piezoresistivity and degrading the performance of the MOSFET devices. Thermal stress can also degrade the saturation current of the transistor down to 30% [93]. This parameter limits the maximum permitted number of TSVs on 3D-NoC by increasing the Keep-Out Zone (KOZ) parameter [63, 67].

- **Thermal concerns:** Thermal concerns and effective heat removal become more critical by stacking and dense packing of circuits on a chip. Temperature cycling and thermal shock accelerate fatigue failures depending on the temperature ranges [48]. Thermomigration is a mass transport driven by a temperature gradient. This can make a homogeneous alloy non-homogeneous under a temperature gradient. It has not been a concern in Al and Cu interconnect technology; but it is now recognized as a serious reliability problem for flip chip solder joints in 3D packaging. This dissertation does not consider thermomigration as a separate source of thermal faults as it deteriorates the effects of other thermal fault sources.

Transistors, contacts, multi-layered Cu, Al interconnects, and solders joints are all the source of heat in 3D designs. The majority of power consumption is attributed to transistors, because the resistivity of silicon is much higher than metals and there are more than a billion transistors on advanced chips. Fur-

thermore, the heat generated by solder joints is much lower than the transistor and interconnects metallization, it affects locally the area surrounding (underfill) solder bumps. The polymer-based underfill has a low glass transition temperature, so the thermal concern may change the viscosity of the underfill. The flow of the underfill reduces its role in the protection of the bump as well as the chip. Thermal concern is considered as waste-heat that increases the conductor temperature, resulting in more joule heating. Thermal concern increases the device temperature and affects atomic diffusion [105]. Four main thermal issues in 3D-NoCs are reported in the following subsections:

### **Electromigration**

Electromigration is a failure mechanism where electrons flowing through metal (Al, Cu) lines collide physically with the metal atoms, causing the metal atoms to migrate and form voids in the metal lines which leads to increased metal line resistance and disconnection. Electromigration is a key failure mechanism that determines the long-term reliability of metal lines. It strongly depends on the material of the metal. Copper has more resistance against electromigration as compared to Aluminum. It also depends on the running temperature of the system [34].

### **Time-Dependent Dielectric Breakdown (TDDB)**

The importance of leakage power has increased dramatically as technology scales down. Leakage (off-state) current has a direct relation with temperature while it has reverse relation with on-state current of a transistor [69]. Additionally, leakage current of MOSFET gates depends on the quality and thickness of oxide

gate. When the leakage current of gate oxide reaches its limitation, the breakdown may happen which results in the failure of device. Furthermore, thin gate oxide and their silicon/silicon-dioxide interface are affected by various physical mechanisms like Hot Carrier Injection (HCI), Negative Bias Temperature Instability (NBTI), and TDDB. TDDB has been considered among researchers as a significant failure mode for deep sub-micron technologies. TDDB is known as one of the main issues for high temperature in thin oxides [69, 75, 65]. It is much more critical in 3D designs because of thermal concerns and thermomigration effect among the layers.

### **Stress migration**

Stress migration is a failure mechanism where stress applied to metal lines causes the metal atoms to creep which forms voids in metal lines. Stress is generated in the metal lines (Al, Cu) used in the IC due to temperature differences between the heat treatment process in the manufacturing process and the operating environment temperature. This stress can cause composition deformation in metal lines, resulting in short-circuits between metal lines, or vacancies in the metal lines causing creep and converge in a single location which consequently form a void [34].

Stress migration occurs due to the interaction between the metal line stress and the metal atom creep speed. Whereas the metal atom creep speed increases at high temperatures, the stress acting on the metal lines decreases at high temperatures, so there is known to be temperatures peaks at which stress migration occurs.

## **Thermal cycling**

Thermal cycling is the process of cycling through two temperature extremes. It causes cyclic strains and develops cracks in a similar way to natural usage and weakens the joint structure by cyclic fatigue [16]. The effect of thermal cycling gets worse in 3D designs, since the temperature of different layers are not the same and there is always a thermal flow between layers. Thermal cycling may also affect other thermal issues such as electromigration, stress migration, and thermal stress.

- **SEE impacts:** SEEs induced by heavy ions, protons, and neutrons become an increasing limitation of the reliability of electronic components, circuits, and systems. SEE has been the main concern in space applications, with potentially serious consequences for the spacecraft, including loss of information, functional failure, or loss of control. It can be destructive or transient, according to the amount of energy deposited by the charged particle and the location of strike on the device. IC malfunctions due to radiation effects from high energy alpha particles at ground level are a major concern because of continued technology size scaling [87]. The electron carriers are collected by the electric field and cause the charge collection to expand resulting in a sudden current pulse. The diffusion current dominates until all the excess carriers have been collected, recombined, or diffused away from PN junction area [46, 35]. In general, SEE is divided into two main categories: Single Event Upset (SEU) and Single Event Transient (SET) [9].

## **SEU**

It is a change of state caused by ions or electromagnetic radiation striking a



sensitive node in a micro-electronic device. These phenomena can affect the behavior of sequential circuits such as memory cells, register files, pipeline flip-flops, and cache memories. The sensitivity of both PMOS and NMOS transistors are high when they are off.

## **SET**

It is a temporary variation in the output voltage or current of a combinational circuit due to the passage of a heavy ion through a sensitive device results in an SET. In analog devices, SETs called Analog Single Event Transients (ASETs), are mainly transient pulses in operational amplifiers, comparators or reference voltage circuits. When a charged particle hits a logic cell sensitive node in the combinational logic, it generates a transient pulse in a gate that may propagate in a path and eventually be latched in a storage cell [52].

### **2.3.2 Logic-level Fault Models**

Logic-level fault models represent the effect of physical faults on the behavior of a modeled system. The results of early studies with logic-level fault models provide the basis for fault simulation, test generation, and other testing analysis applications.

A higher level fault model allows derivation of the inputs that test the chip without knowing any of these physical details of failures. Logical or electrical malfunction of the target system can be estimated by applying fault models. However, there is a tradeoff between modeling accuracy and simulation time in physical and logic-level fault modeling. The accuracy of logic-level fault modeling would increase by covering

Table 2.1: Physical faults and their corresponding logic-level fault models

Physical Fault	Cause	Fault Model
Chip warpage	✓ Crack in Micro-bumps [105]	✓ Open-circuit
TSV coupling	✓ Increase path delay due to Miller effect [57, 62] ✓ May result in a wrong logic function switch and cause unintentional flip in the target signal	✓ Delay-fault ✓ Bridging
Thermal stress	✓ Affects carrier mobility of transistor which degrades the performance [90]	✓ Delay-fault
Electromigration	✓ Increase wire resistance [34] ✓ Disconnection	✓ Delay-fault ✓ Open-circuit
TDDB	✓ Dielectric breakdown of the gate dielectric film influences on transistor behavior	✓ Stuck-at-0-1
Stress migration	✓ It deforms composition in metal lines or makes vacancies (atom holes) in the metal lines, forming a void by creeping and converging in a single location [75]	✓ Open-circuit ✓ Short-circuit
Thermal cycling	✓ Interfacial crack [93]	✓ Open-circuit ✓ Delay-fault
SEU	✓ Electrical noise induced by high energy ionizing particles on sequential circuits	✓ Bit-flip
SET	✓ Electrical noise induced by high energy ionizing particles on combinational circuits	✓ Stuck-open

all the necessary areas of testing analysis. On the other hand, physical-level is not adapted for the simulation of complex designs such as 3D-NoCs because of their long simulation time. Modeling the physical fault effects by the logic-level faults is preferable, in order to keep the complexity low and simulation time short, while meeting the required accuracy [32, 37]. This approach decreases the complexity by employing a

single logic-level fault model for different technologies, and representing the effects of physical faults which are not completely understood. This approach is the fundamental concept for methods presented in Chapter 4.

Additionally, logic-level model of 3D-NoC for the qualification test must be close enough to the physical-level to include all the unwanted impacts of existing physical faults on the system. Understanding causes and effects of physical faults prevents extreme computation time augmentation in logic-level modeling by providing an intelligent fault modeling technique. Table 2.1 summarizes causes and correlated logic-level fault models for each of the introduced physical faults. The detailed description of these logic-level faults are presented in [13, 106, 85, 18, 86, 99].

## 2.4 Fault Effects on 3D-NoC

3D-NoC is composed of different components and links, including connected 2D planar structure by inter-die connections. The vulnerable elements of 2D planar NoC and TSV links are itemized in Figure 2.3. Categorizing sensitive components of 3D-NoC results in more accurate and faster reliability analysis. Different sources of fault may have different effects in 3D-NoC. Figure 2.3 also depicts the effects of potential faults on main components of 3D-NoC. These effects include header/data flit loss, packet drop, packet truncation, packet latency, misrouting, timing jitter, flit corruption, and disconnection. Reporting the percentage of each of these metrics is a fruitful approach to compare the reliability of any fault-tolerant design. A brief description of them is listed below:

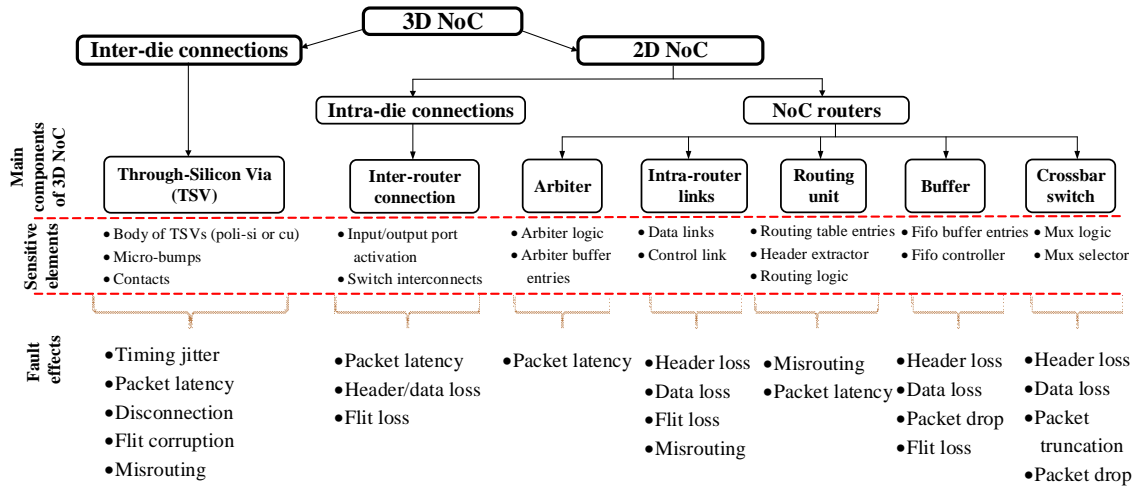


Figure 2.3: Classification of faults and their effects in 3D-NoC components

- Header/data flit loss:** It represents any header/data flit alteration of an incoming packet. It would happen if there is any problem with the logic circuit of crossbar switches, FIFO controller, or internal connections among components inside a router. Header flit loss which has a lower probability than data flit loss is more critical. A packet may never reach its expected destination, if a header loss occurs.
- Packet drop:** The incoming packet will be skipped or forwarded to an invalid output port if a fault occurs in input buffer pointers/counters or output switch.
- Packet truncation:** It happens when erroneously a data flit is recognized as a trailer flit, where one or more data flits from the end of a packet will be lost.
- Packet latency:** Packet latency is the result of the router arbitration malfunction because of a fault inside the arbiter logic, internal connections, or TSV connections. TSV issues and some of the thermal related faults may change the conductivity of TSVs or internal links by changing their physical structures.

- **Misrouting:** Misrouting is the consequence of fault occurrence on either header flit while transferring among NoC architecture components or on comparator modules of routing unit component. It can be resolved if it happens because of any transient faults. Permanent faults in a routing unit component result in extra number of packet transmissions between adjacent routers and consequently causing congestion in some parts of the network.
- **Timing jitter:** The short-term variations of a digital signals significant instants from their ideal positions in time is called timing jitter. It is a significant, and usually undesired, factor in the design of almost all communications links. The delay caused by temperature variation and TSV coupling, leads to a timing jitter, which is more significant for TSV because of its size.
- **Flit corruption:** Flit corruption in a packet happens when a fault occurs in the data path of a router. It may occur in intra-router links, crossbar switch components, or TSV link.
- **Disconnection:** Different types of faults may cause the vertical link components (TSV body, contact, micro-bump) to fail. Electromigration affects the body and micro-bumps, causing the path to be implied as an open link or disconnected.

## 2.5 Failure Analysis of 3D-NoC

This section provides an analytical method to estimate the failure probability caused by different sources of physical faults in a 3D-NoC to save design and verification time. First, a formal failure analysis for TSV coupling issue is provided. Then, thermal

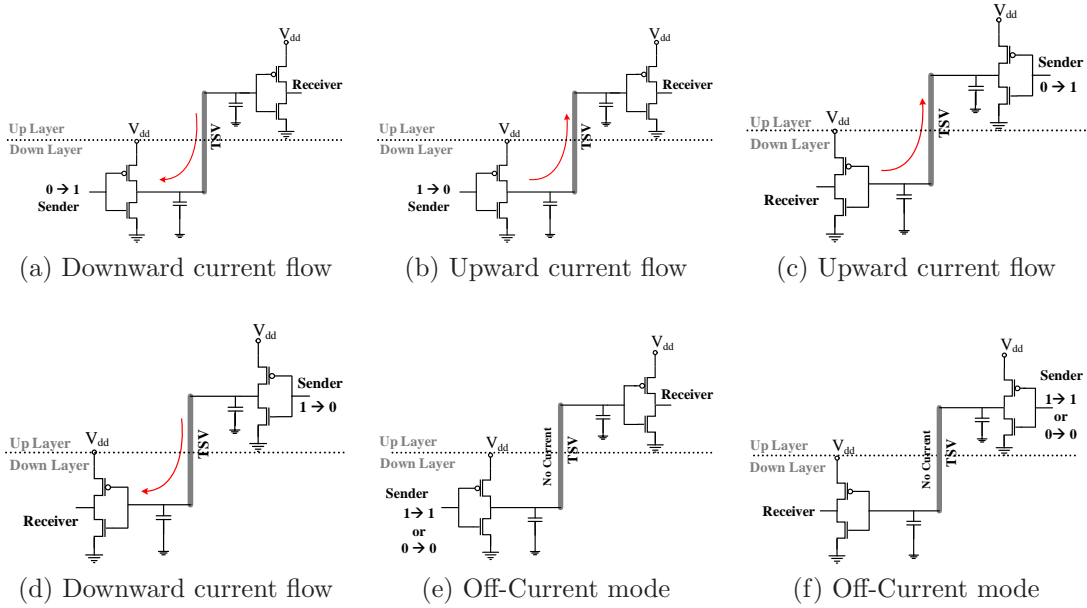


Figure 2.4: Current flow direction in TSV.

issue failure evaluation for different parts of the 3D-NoC router is reported in terms of Mean-Time to Failure (MTTF). Finally, the benefit of stacked structure of the 3D-NoC against SEE physical faults is highlighted.

### 2.5.1 TSV Issues

Failure analysis of TSV caused by thermal stress needs a full system simulation to profile the chemical interaction of materials with different CTEs. Furthermore, chip warpage mostly happen during manufacturing process [105, 66]. Thus, both of them are outside the scope of this dissertation. However, TSV coupling analysis is discussed in this dissertation, when at least two neighboring TSVs have flow of electric charge. This concept will be used later in Chapters 4, 5, 6, and 7 to explain the proposed methods.

The current flow direction of a TSV is data-dependent, based on charging and discharging of the intermediate capacitor between each pair of transistors of stacked layers. The behavior of the intermediate capacitor relies on the transmitted and ready to transmit data bit values. Figure 2.4 illustrates six possible cases depending on the data bit values and location of the sender, resulting in three possible current flows in TSVs. There is a downward current flow when the input data bit of sender changes from '0' to '1' and '1' to '0' if the sender is in lower and upper level, respectively, as shown in Figure 2.4a and Figure 2.4d. Similarly, there is an upward current flow direction, if the data bit of the sender changes from '1' to '0' and '0' to '1' if the sender is in lower and upper level, respectively, as shown in Figure 2.4b and Figure 2.4c. TSV does not carry any current, if there is no switching between the transmitted and ready to transmit data bit, as shown in Figure 2.4e and Figure 2.4f. In the rest of this dissertation, such a TSV is called an inactive TSV, which does not have any current flow. The  $\odot$ ,  $\otimes$ , and  $\circ$  symbols represent the upward, downward current flow directions, and inactive TSV, respectively.

Figure 2.5 shows the cross section view of TSVs in 3D-NoC. Throughout this dissertation, unless mentioned otherwise, it is assumed that only adjacent TSVs (both vertical or horizontal direction) have mutual coupling effect on each other. With this assumption a TSV coupling may happen in four different cases based on the number and arrangement of active neighbors for a victim TSV. The colored and gray TSVs, shown in Figure 2.5, represent victim and aggressor TSVs, respectively. An inductance and capacitive parasitic categorization and formal reliability analysis for both of them are provided in the following subsections.

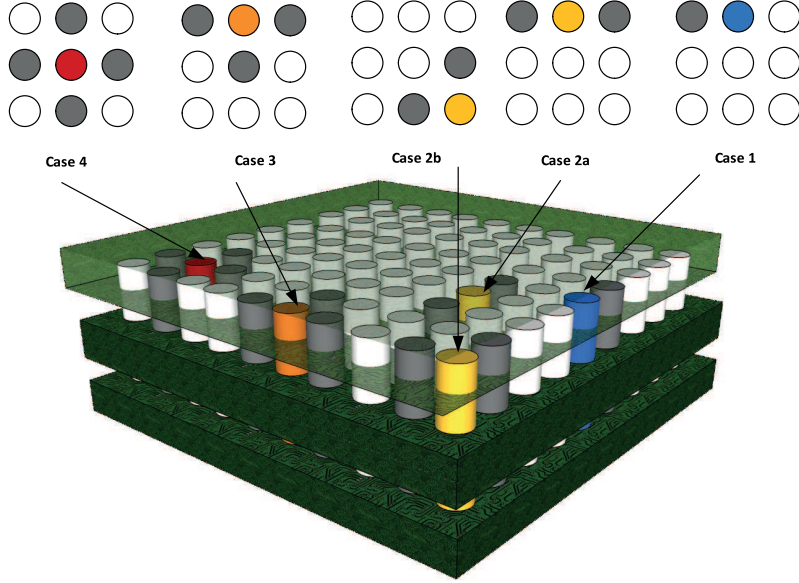


Figure 2.5: Different TSV patterns leading to coupling

### Inductive TSV Coupling

The inductively-induced coupling failure probability of a TSV is a function of the unexpected total coupled voltage ( $V_{Icoupl_{tot}}$ ) caused by its active neighbor TSVs.

If the inductive coupling voltage caused by a single horizontal or vertical neighboring TSV is  $\beta$ ; then total inductively coupled voltage on a victim TSV ( $V_{Icoupl_{tot}}$ ) is proportional to  $\alpha\beta$ , where the value of parameter  $\alpha$  depends on the current flow direction and arrangement of active neighboring TSVs. Assuming the electromagnetic proximity effect and other high order effects are negligible, the  $V_{Icoupl_{tot}}$  is equal to sum of the voltages induced by each aggressor TSV, based on Faraday's law as shown in Equation 2.1.

$$V_{Icoupl_{tot}} = \sum_{i=1}^N V_{Icoupl_i} = \sum_{i=1}^N M_{v,i} \frac{dI_i}{dt} \sim \alpha\beta \quad (2.1)$$



where  $N$  is the total number of aggressors,  $V_{Icoupl_i}$  is the voltage coupled on the victim by  $i^{th}$  aggressor, assuming all other aggressors have constant current.  $M_{v,i}$  is the mutual inductance between  $i^{th}$  aggressor and victim TSVs.  $I_i$  is the current of  $i^{th}$  aggressor TSV. The  $M_{v,i}$  is extracted from Equation 2.2 [103].

$$M_{v,i} = \frac{\mu_0}{2\pi} \left[ l \ln \left( \frac{l + \sqrt{d_i^2 + l^2}}{d_i} \right) + d_i - \sqrt{d_i^2 + l^2} \right] \quad (2.2)$$

where  $d_i$  is the distance of  $i^{th}$  aggressor from the victim TSV and  $l$  is the TSV length. It can be concluded from Equation 2.1 that an inactive TSV does not have inductive coupling effect on a victim TSV. In other words, a victim TSV on the border line of TSV configuration (case 3 in Figure 2.5) has the same probability of failure with the victim one in the middle of TSV configuration (case 4 in Figure 2.5) which has one inactive neighbor. With this point of view, each victim TSV has four active/inactive neighbors in horizontal or vertical directions. Therefore, there are  $3^4 = 81$  possible configurations of TSV neighbors for a victim TSV, according to the current flow of its neighbors. Many of these 81 arrangements behave similarly as long as the  $\alpha$  is concerned (see Table 2.2). The  $\oplus$  symbol is a victim TSV regardless of its current direction which does not impact our proposed analysis. The first column shows the corresponding case number of neighboring TSV arrangements in Figure 2.5. In other words, the case number is equal to the number of active neighboring TSVs. Both of *case2a* and *case2b* of Figure 2.5 have similar effect on the victim TSV; they are merged in Table 2.2. Five distinct absolute values for  $\alpha$  are reported for each of the possible patterns in the second column of Table 2.2. The sign of  $\alpha$  (negative or positive) is determined by the direction of its neighbor TSVs' current flow direction. The effect of both positive and negative  $\alpha$  for the same absolute value is symmetric. This is the

Table 2.2: TSV-to-TSV inductive coupling categorization

Case	$ \alpha $	Sample Patterns	Occurrence frequency
—	0	$\oplus$	1
1	1	$\oplus \odot$	8
2	2	$\odot \oplus \odot$	12
	0	$\odot \oplus \otimes$	12
3	3	$\odot$ $\odot \oplus \odot$	8
	1	$\odot$ $\odot \oplus \otimes$	24
4	4	$\odot$ $\odot \oplus \odot$ $\odot$	2
	0	$\otimes$ $\odot \oplus \odot$ $\otimes$	6
	2	$\odot$ $\odot \oplus \odot$ $\otimes$	8

reason that only absolute factors of  $\alpha$  values are reported in Table 2.2. A sample pattern of TSV current flow configuration and their occurrence frequency are presented in the third and fourth columns of Table 2.2, respectively. If the inductive coupling voltage by a single neighbor on the victim has a magnitude of  $\beta$ , the corresponding failure probability is denoted by  $P_\beta$ . Similarly,  $P_{\beta_{\text{tot}}} = P_{\alpha\beta}$  represents the total failure probability on a victim for a given TSV neighboring configuration because of inductive coupling noise. TSV failure probability caused by inductive coupling ( $P_{f_{ind}}$ ) is calculated by integrating the occurrence frequency of each of the  $\alpha$  values with their corresponding failure probabilities, as shown in Equation 2.3.

$$P_{f_{ind}} = \frac{1}{81} [19P_{0\beta} + 32P_{1|\beta} + 20P_{2|\beta} + 8P_{3|\beta} + 2P_{4|\beta}] \quad (2.3)$$

Table 2.3: TSV-to-TSV capacitive coupling categorization

$TC_{tot}$	0C	1C	2C	3C	4C	5C	6C	7C	8C
SM <sup>1</sup>	⊙ ⊙ ⊙ ⊙ ⊙	⊙ ⊙ ⊙ ⊙ ○	⊙ ⊙ ⊙ ⊙ ⊗	⊙ ⊙ ⊙ ○ ⊗	⊙ ⊙ ⊙ ⊗ ⊗	○ ⊙ ⊙ ⊗ ⊗	⊗ ⊙ ⊙ ⊗ ⊗	⊗ ○ ⊙ ⊗ ⊗	⊗ ⊗ ⊙ ⊗ ⊗
OF <sup>2</sup>	3	16	44	64	54	32	20	8	2
OP <sup>3</sup>	0.01	0.07	0.18	0.26	0.22	0.13	0.08	0.03	0.01

in which,  $P_{0\beta} < P_{1|\beta} < P_{2|\beta} < P_{3|\beta} < P_{4|\beta}$ . Although, the  $P_{\alpha\beta}$  is introduced here as a failure probability caused by induced coupling voltage, it will be equal to 0 or 1 for a fixed physical parameters and operational frequencies. But it can be still considered as an evaluation metric in order to compare the efficacy of any proposed fault-tolerant approach, as discussed in [28, 112].

### Capacitive TSV Coupling

The parasitic capacitance on the TSV has a wide slow-varying nature which noticeably affects the timing requirements of the circuit operation. It certainly slows down transitions on a switching signal when victim TSVs' neighbors perform opposite transitions. Generating large glitches on a static signal is another unexpected effect for a victim TSV when its aggressive neighbors have opposite transitions. The capacitive coupling value is a function of the charging and discharging of the victim TSV and its neighbors due to the Miller effect [58]. Since between any two dies, the TSV drivers are all in one die and the loads are in the other die, the direction of the current flow in each TSV can specify the charging or discharging state. On the other hand, the

---

<sup>1</sup>Sample Pattern

<sup>2</sup>Occurrence frequency

<sup>3</sup>Occurrence probability

current flow direction is data-dependent as discussed earlier.

However, contrary to inductive TSV coupling, an inactive TSV is important in categorizing capacitive coupling. With this consideration, the calculation of capacitive coupling for a victim TSV on the border line (case 1, case 2, and case 3 in Figure 2.5) and in the middle of TSV array (case 4 in Figure 2.5) with three, two, or one inactive neighboring TSV are not the same. However, only case 4 of Figure 2.5 is considered here to categorize the effects of capacitive coupling, because of the following reasons. First, since the number of TSVs in the middle of TSV array is more than the ones on the border. Second, the probability of a TSV failure caused by capacitive coupling with more neighbors, is higher than the failure probability of a TSV on the border. Third, similar calculation for capacitive coupling failure can be done for all the remaining cases in Figure 2.5 with the similar method, since their patterns are a subset of existing patterns in case 4.

With these assumptions, the parasitic capacitance value between a pair of TSVs is represented by 0C (if they both have the same current direction), 1C (if one of them is inactive and the other is active), 2C (if they have reverse current flow). The total capacitive coupling noise for a victim TSV is equal to the sum of coupled voltages by each aggressor on the victim TSV. If we represent an upward current with +1, a downward current with -1 and no-current with 0, the total capacitive coupling noise for a victim TSV can be quantified using Equation 2.4.

$$TC_{tot} = \sum_{i=1}^N |I_{vic} - I_{agg_i}| \quad (2.4)$$

where  $TC_{tot}$  is the total capacitance coupling noise on a victim TSV and N represents

the number of its adjacent aggressors. Consequently, according to Equation 2.4 the maximum value for  $TC_{tot}$  is 8C.

Considering five TSVs ( four aggressors and a victim) and three possible different values for each (upward, downward, no current), there are  $3^5 = 243$  possible TSV coupling patterns. Many of these 243 arrangements are similar as long as the severity of capacitive coupling is concerned. Table 2.3 categorizes the TSV capacitive coupling in terms of their parasitic capacitance values. A sample pattern of current flow in TSVs with their occurrence frequency and probability for each of these parasitic values are also shown in this table.

Similar to the discussion in Section 2.5.1, TSV failure probability caused by capacitive coupling ( $P_{f_{cap}}$ ) is calculated by integrating the frequency occurrence for each  $TC_{tot}$  values, as shown in Equation 2.5.

$$P_{f_{cap}} = \frac{1}{243}[3P_{0C}+16P_{1C}+44P_{2C}+64P_{3C}+54P_{4C}+32P_{5C}+20P_{6C}+8P_{7C}+2P_{8C}] \quad (2.5)$$

in which,  $P_{0C} < P_{1C} < P_{2C} < P_{3C} < P_{4C} < P_{5C} < P_{6C} < P_{7C} < P_{8C}$ .

## 2.5.2 Thermal Concern

Thermal effects are considered as one of the main challenges for future design. It has become more critical by the emergence of 3D designs. An MTTF for the main components of a given system is a good approximation to estimate its failure rate [34] which is defined for a non-repairable component. In other words, MTTF provides the expected life time of a component. Failure probability of different components is

a function of MTTF. It might depend on temperature, material characteristics, and current flow, which can be modeled by Weibull distribution (as shown in Equation 2.6).

$$P_f(t) = 1 - \exp^{-(\lambda t)^\beta}, \quad \lambda = \frac{1}{MTTF} \quad (2.6)$$

The Weibull parameter  $\beta$  signifies the rate of failure. The MTTF of major components in a 3D-NoC, evaluated based on thermal concerns, are explored in following subsections.

## **Electromigration**

Electromigration has been one of the major reliability problems in conventional 2D designs. It also affects TSV links among silicon layers. Electromigration effect depends on geometrical shapes of wires, temperature distribution, mechanical stress, current density, and material properties. It modifies the expected connectivity of components by generating voids and hillocks in wires and links. As the electromigration-induced system lifetime is inversely proportional to the square of current density, small increase in current density can decrease the system's lifetime significantly.

While some advancements have been made to reduce the effects of current crowding, electromigration can increase the dissolution of Cu into solder and lead to failure. It affects any communication links, including wires in 2D designs and TSVs. MTTF of wire links and TSVs because of electromigration is commonly described by Black's

model, as shown in Equation 2.7 [97].

$$MTTF_{EM} = \frac{A_{EM}}{J^n} e^{-\frac{E_{\alpha EM}}{kT}} \quad (2.7)$$

where  $A_{EM}$  is a constant determined by the physical characteristics of the metal interconnect,  $J$  is the current density,  $E_{EM}$  is the activation energy of electromigration,  $n$  is an empirically-determined constant, and  $T$  is the temperature.

## TDDB

Gate oxide reliability is most certainly the principal concern in modern microelectronics as the CMOS device dimensions continuously scale down. TDDB refers to an important failure process in VLSI designs and strongly depends on the system temperature. It is a function of the thickness of gate-oxide and value of supply voltage which gets worse as technology size scales down. It influences the reliability of 2D design, containing billions of transistors. MTTF of transistor gates impacted by the TDDB effect can be calculated by Equation 2.8 [13].

$$MTTF_{TDDB} = A_{TDDB} \left( \frac{1}{V} \right)^{(a-bt)} e^{-\frac{A + \frac{B}{T} + CT}{kT}} \quad (2.8)$$

where  $A_{TDDB}$  is a constant,  $V$  is the supply voltage, and  $a$ ,  $b$ ,  $A$ ,  $B$  and  $C$  are fitting parameters.

## Stress migration

The term stress migration describes the movement of metal atoms under the influence of mechanical stress gradients. Generally, stress gradients can be assumed to be proportional to the applied mechanical stress. Little metal movement (migration) occurs until stress exceeds the yield-point of the metallization [13]. Equation 2.9 shows how to obtain MTTF of different metal layers in 2D design.

$$MTTF_{SM} = A_{SM} |T_0 - T|^{-n} e^{\frac{E_{\alpha SM}}{kT}} \quad (2.9)$$

where  $A_{SM}$  is a constant,  $T_0$  is the metal deposition temperature during fabrication,  $T$  is the run-time temperature of the metal layer,  $n$  is an empirically-determined constant, and  $E_{SM}$  is the activation energy for stress migration.

## Thermal cycling

Thermal cycling happens because of thermal mismatch in adjacent material layers with different coefficients. In chip and package, adjacent material layers, such as copper and *low-k* dielectric, and in 3D architectures between TSV body material, micro-bump, and silicon have different coefficients of thermal expansion. As a result, run-time thermal variation causes fatigue deformation, leading to failures. The MTTF due to thermal cycling is given by the Equation 2.10 [13].

$$MTTF_{TC} = \frac{A_{TC}}{(T_{average} - T_{ambient})^q} \quad (2.10)$$



where  $A_{TC}$  is a constant coefficient,  $T_{average}$  is the chip average run-time temperature,  $T_{ambient}$  is the ambient temperature, and  $q$  is the Coffin-Manson exponent constant.

### 2.5.3 SEE Impacts

SEE caused by Electromagnetic Interferences (EMI), alpha particles strikes, or cosmic radiations have been considered as major sources of faults for electronic circuits. SEE does not have any conspicuous influence on TSV links, while it can modify the charge of transistors in a 2D design. The Soft Error Rate (SER) in different silicon layers of 3D-NoC is not the same. It is necessary to find the flux of incoming particles toward transistors on each silicon layer and calculate the amount of charge deposited by each of them to evaluate SER in different silicon layers.

Failure probability of SEE cannot be calculated with the same method as failure probability of thermal issues. SEE effect has transient impact on the target system. MTTF is defined for non-repairable systems as described earlier; for SEE effect the term Mean Time Between Failure (MTBF) is defined. However, MTBF cannot be used in the same way as MTTF to calculate the failure rate of a component [57].

It is proven that in 3D designs the outer die behaves as a shield and protects inner dies against striking particles by reducing their energy and flux. It is reported that outer dies of 3D design can stop more than 90% of incident alpha particles. The thick bulk silicon is also able to stop majority of striking particles [123]. In other words, SEE has similar influence on 3D-NoC as it only targets the first silicon layer. Many researches have studied SEU and SET impacts on 2D-NoCs [32, 23]; thus, this chapter does not provide an specific reliability analysis of SEE.

The shielding property of 3D ICs provides more opportunities for optimizing soft error tolerant techniques. In other words, vulnerable components can be mapped to inner silicon layers which are more robust. However, increasing the circuit density is limited due to thermal issues in inner silicon layers.

## 2.6 Summary

A taxonomy of potential physical faults and their corresponding logic-level fault models were presented to support an accurate simulation and time-efficient profiling for experimental reliability assessment methods. Such taxonomy leads to more methodical formal analysis. It evaluates those components that are more susceptible to failure for a 3D-NoC design against potential physical faults, which is arguably different from 2D designs. MTTF of components caused by the majority of physical faults is a function of temperature while different active silicon layers of 3D design do not have the same temperature and behavior against the physical faults. It is also explained that SEE faults affect only the outer layer of 3D-NoC designs, since the majority of alpha particles are stopped by the outer layer and they cannot reach the other layers. Finally, accurate failure analysis for major sources of faults for an operational 3D-NoC design is discussed.

# Chapter 3

## Formal Reliability Analysis of 3D-NoC

### 3.1 Introduction

Reliability is a significant challenge for 3D chip designers as exascale computational performance comes to fruition, in the near future [14, 121]. Furthermore, increasing power consumption, power variation, and power density have negative impacts on reliability of 3D chip designs. Rapid changes in power consumption uncovers voltage fluctuations, leading to more frequent transient errors. On the other hand, high temperatures increase leakage power consumption, resulting in a self-reinforcing cycle of dependency between power and temperature [69, p. 25]. Power consumption and consequently temperature have a direct relationship with the reliability of systems [16].

Reliability of a system can be measured by means of experimental and analytical

methods. An analytical model for reliability evaluation of 2D NoC has been reported in [23], but it does not consider unexpected sources of faults in 3D die-stacked designs. Accuracy of simulation environment is a concern for experimental methods in order to analyze reliability of a system. Furthermore, measurements are expensive and time-consuming while time-to-market cycle is of great importance. A full-chip thermo-mechanical stress and reliability analysis tool and a design optimization methodology have been presented to alleviate mechanical reliability issues in 3D integrated circuits (ICs) [47]. Reliability evaluation of a specific TSV technology developed by Austria Microsystems AG has been reported in [17]. Although, many researches have focused on reliability issues for 3D-NoC architectures, developing general analytical techniques to advance both the intuitive understanding and the quantitative measurement of how potential physical faults influence the behavior of 3D NoC are lacking. This chapter focuses on TSV technology as one of the most promising technologies for 3D IC integration and aims to provide a formal analysis of a 3D-NoC reliability under TSV failure.

An analytical model for reliability evaluation of 2D NoC has been reported in [23], but it does not inspect thermal effects which are critical for 3D die-stacking designs. There are still additional issues for developing 3D architecture EDA tools [15].

In this chapter, a TSV-based 3D-NoC is modeled mathematically. Several assumptions are made during the modeling to facilitate finding a closed form expression for system reliability. Then, system reliability is defined quantitatively. We find a mathematical expression for system reliability that depends on two scalar parameters, namely injection rate and TSV failure probability, and a simulation-based factor that depends only on network architecture. A significant contribution of this chapter is

that only one simulation run is required to estimate system reliability for all values of injection rate and TSV failure probability; This is in contrast with the traditional approach in which a full simulation run is required for each value of injection rate and TSV failure probability. With the proposed approach, it is possible to efficiently estimate system reliability for a wide range of system parameters by simply changing the parameters of final expression. For systems that do not strictly obey the assumptions, but do not deviate significantly, it is possible to use this approach to estimate the *order of magnitude* of system reliability.

The rest of this chapter is organized as follows: Reliability analysis of TSV-based NoC is discussed in Section 3.2, and Section 3.3 delivers some conclusion remarks.

## 3.2 Reliability Analysis

Having discussed the sources of TSV failure in Chapter 2, the next step is to inspect the effect of these TSV failures on the reliability of a 3D NoC using TSV for vertical integration.

Consider an NoC consisting of IPs connected to routers, and vertically adjacent routers interconnected by TSVs. The general goal of the system is to transfer data between different IPs (routers). Assuming a time-slot based operation for the system, a subset of TSVs will be selected to participate in data transfer at each time slot. Parameters such as location and number of source-destination router pairs, buffering strategy and buffer status at routers, and routing algorithm affect the selection of participating TSVs at each time step.

After modeling the source-destination pairs in Section 3.2.1, a reliability criterion, namely *probability of system failure*, for the operation of NoC during one time slot is defined. Then, the relation between *probability of system failure* and parameters such as *traffic injection rate* and *TSV failure probability* is quantified. The equation relating these parameters has a convenient analytical expression with only one term, which requires running an architecture dependent simulation.

### 3.2.1 Source-Destination Status

Consider an arbitrary 3D array of  $N_r$  routers communicating with each other during a specific time slot  $t$ , where  $t$  is an integer indexing the time slot. NoC routers are indexed by  $n_r = 1, \dots, N_r$ . At time slot  $t$ , depending on the traffic model, some of the routers have data to send; such routers are called data sources. The index of these data sources in time slot  $t$  is represented by an  $N_r \times 1$  vector  $\mathbf{s}^{(t)}$  defined as follows. If the  $n_r^{th}$  router has data to send at time  $t$ , the corresponding element of  $\mathbf{s}^{(t)}$ , is set to 1.

For each source node, the index of the corresponding destination node can be represented using an  $N_r \times 1$  vector. Note that while the elements of  $\mathbf{s}^{(t)}$  are either 0 or 1, the elements of  $\mathbf{d}^{(t)}$  are either 0 or equal to the index of destination router. Furthermore, the proposed definition of  $\mathbf{d}^{(t)}$  implies that the construction of  $\mathbf{d}^{(t)}$  requires knowledge of  $\mathbf{s}^{(t)}$ .

The selection of active TSVs in each time slot is determined by source-destination locations as well as routing algorithm and buffering mechanism. The overall effect of these parameters is reflected in the injection rate of routers. In order to decouple the analysis from the architectural details, the proposed algorithm is designed to be

sensitive to injection rate. The communication network is completely characterized by the pair  $c \triangleq (\mathbf{s}^{(t)}, \mathbf{d}^{(t)})$  satisfying constraint  $\mathfrak{C}$ . Let's denote the set of all such  $c$ 's by  $\mathcal{C}$ . This matrix is extracted per application from the architectural simulation. The cardinality of this set is shown by  $|\mathcal{C}|$  and the members are indexed by  $i = 1, 2, \dots, |\mathcal{C}|$ , and  $c_i = (\mathbf{s}_i, \mathbf{d}_i)$  represents the  $i_{th}$  element of set  $\mathcal{C}$ . Since each  $c_i$  contains all the information about the NoC communication configuration,  $c_i$  is called as ‘‘communication configuration.’’

### 3.2.2 TSV Status

At the beginning of each time slot, some IPs inject data to the connected routers to be transferred to other IPs. The data link among the pairs of routers are established by an arbitrary 3D array of  $N_t$  TSV bundles. Each TSV bundle is responsible for connecting vertically adjacent routers, as illustrated in Figure 2.1. In this chapter, the term ‘‘TSV bundle’’ is used interchangeably with ‘‘TSV.’’ The TSVs are indexed by  $n_t = 1, \dots, N_t$ . Based on the vectors  $\mathbf{d}^{(t)}$  and  $\mathbf{s}^{(t)}$  and the routing algorithm, a set of TSVs are chosen to enable the communication between routers at time slot  $t$ . The status of TSVs at time slot  $t$  is represented by an  $N_t \times 1$  vector  $\mathbf{t}^{(t)}$ . The selection process is abstracted as a mathematical operator  $\mathcal{T}$ , defined as the operator taking  $\mathbf{d}^{(t)}$  and  $\mathbf{s}^{(t)}$  as input and returning  $\mathbf{t}^{(t)}$  as the output.

### 3.2.3 Reliability Analysis

The *probability of system failure* for a given communication configuration  $c_i$  is defined as the probability that at least one of the engaged TSVs supporting  $c_i$  fails in a given time slot. Assume that the effect of all discussed TSV issues can be modeled by a time-invariant constant  $P_{f,TSV}$ , designating the probability of failure of an engaged TSV.  $P_{f,TSV}$  is also assumed to be the same for all TSVs. This assumption holds if:

- Data generation is spatially uniform at each time slot, i.e. all IPs follow the same data generation behavior and are not biased toward a specific destination IP. Uniform traffic generation has been used in the literature to analyze the performance of NoC [8]. Uniform data pattern also results in temporally and spatially uniform coupling.
- Data generation is also temporally uniform or at least uniform over some observation interval of interest. This means that data generation behavior does not change over time.
- Chip temperature has reached steady state. This holds if the system has been working for a minimum amount of time and if data generation is both spatially and temporally uniform.

The probability of system failure for a given communication configuration  $c_i$  is equal to complement of the probability of having no faulty TSVs:

$$P_f(c_i) = 1 - [1 - P_{f,TSV}]^{||\mathcal{T}[c_i]||} \quad (3.1)$$



where the operator  $\|\cdot\|$  returns the sum of elements of its input argument. Consequently,  $\|\mathcal{T}[c_i]\|$  can be interpreted as the number of active TSVs supporting the communication configuration  $c_i$ . Furthermore, Equation 3.1 implicitly assumes that failure of engaged TSVs are independent. This assumption is true when:

- The distance between TSV bundles, which is the same as the distance between routers, is long enough such that TSV coupling becomes a local issue (within a bundle, not between bundles) and
- Temperature distribution is spatially uniform

To calculate the total failure probability, a weighted average of failure probabilities of all  $c_i$ 's should be performed. Since different communication configurations (different members of  $\mathcal{C}$ ) occur with different probabilities, weights of the mentioned averaging are probabilities of occurrence of  $c_i$ s. Denoting the probability of occurrence of communication configuration  $c_i$  by  $P_o(c_i)$ , the total probability of system failure is:

$$P_{f,\text{system}} = \sum_{i=1}^{|\mathcal{C}|} P_o(c_i)P_f(c_i) \quad (3.2)$$

Obviously,  $P_{f,\text{system}}$  is less than 1 because  $\sum_{i=1}^{|\mathcal{C}|} P_o(c_i) = 1$  and  $P_f(c_i) < 1$  for all  $c_i \in \mathcal{C}$ .

The remaining task is to calculate  $P_o(c_i)$ . To calculate  $P_o(c_i)$ , rewrite it as:

$$P_o(c_i) = \Pr(\mathbf{s} = \mathbf{s}_i, \mathbf{d} = \mathbf{d}_i) = \Pr(\mathbf{s} = \mathbf{s}_i) \times \Pr(\mathbf{d} = \mathbf{d}_i | \mathbf{s} = \mathbf{s}_i) \quad (3.3)$$

The first term,  $\Pr(\mathbf{s} = \mathbf{s}_i)$ , is governed by the statistical behavior of routers' data transmission. As mentioned before, it is assumed that all routers have the same statistical

transmission behavior. Furthermore, several factors such as data size distribution, buffer state, and routing algorithm can affect the statistical transmission behavior of routers. It follows from uniform data pattern that these factors can be combined together and modeled by the concept of time-invariant *injection rate*. The injection rate,  $\alpha$ , is defined as the probability that a router has data to transmit during time slot  $t$ . For *time-invariant* injection rate,  $\alpha$  is assumed to be independent of  $t$ .

Using these assumptions, it is easy to relate  $\Pr(\mathbf{s} = \mathbf{s}_i)$  to the number of sources, i.e.  $\|\mathbf{s}_i\|$ , as:

$$\Pr(\mathbf{s} = \mathbf{s}_i) = \alpha^{\|\mathbf{s}_i\|} (1 - \alpha)^{N_r - \|\mathbf{s}_i\|} \quad (3.4)$$

To calculate the second term in Equation 3.3, note that for a given  $\mathbf{s}_i$ , there is more than one  $\mathbf{d}$  for which  $\mathbf{s}_i = \text{sign}(\mathbf{d})$ . This is because knowledge of locations of the sources ( $\mathbf{s}_i$ ) by itself does not fully characterize the communication configuration of system. In other words, all communication configurations for which sources are in the same location, but differ in destination location, have the same  $\mathbf{s}$  vector.

Following the previous assumption of uniform data generation, all possible vectors,  $\mathbf{d}$ , for which  $\text{sign}(\mathbf{d}) = \mathbf{s}_i$ , are equally likely. Consequently,  $\Pr(\mathbf{d} = \mathbf{d}_i | \mathbf{s} = \mathbf{s}_i)$  is simply the inverse of  $\mathcal{M}(\mathbf{s}_i)$ , where  $\mathcal{M}(\mathbf{s}_i)$  is the number of possible vectors  $\mathbf{d}$ , for which  $\mathbf{s}_i = \text{sign}(\mathbf{d})$ :

$$\Pr(\mathbf{d} = \mathbf{d}_i | \mathbf{s} = \mathbf{s}_i) = \frac{1}{\mathcal{M}(\mathbf{s}_i)} \quad (3.5)$$

Since each source can send data to any of the  $N_r - 1$  destinations (all nodes except for itself), and there are a total of  $\|\mathbf{s}_i\|$  sources for given  $\mathbf{s}_i$ ,  $\mathcal{M}(\mathbf{s}_i)$  is equal to:

$$\mathcal{M}(\mathbf{s}_i) = [N_r - 1]^{\|\mathbf{s}_i\|} \quad (3.6)$$

At this point, the summand of Equation 3.2 is calculated by generating all possible members of  $\mathcal{C}$ .

$$P_{f,\text{system}} = \sum_{s=0}^{N_r} \sum_{t=0}^{N_t} \left[ N(s, t) \times \frac{\alpha^s (1 - \alpha)^{N_r - s}}{(N_r - 1)^s} [1 - (1 - P_{f,\text{TSV}})^t] \right] \quad (3.7)$$

where  $N(s, t)$  is the number of communication configurations with  $s$  active sources and  $t$  engaged TSVs. Direct calculation of  $N(s, t)$  requires generating all communication configurations. An alternative approach to calculating  $N(s, t)$  is the use of combinatorial techniques. However, this approach also turns out to be intractable. An indirect approach is proposed for calculating  $P_{f,\text{system}}$ . The proposed approach leads to a simple algorithm as well as an approximate method for finding  $P_{f,\text{system}}$ .

An intuitive interpretation for the problematic terms in Equation 3.7 provides tractable and approximate methods for finding  $P_{f,\text{system}}$ . A more thorough evaluation of Equation 3.7 reveals that both terms  $N(s, t)$  and  $(N_r - 1)^s$  have the same interpretation as they both count occurrences. Quantitatively,  $N(s, t)$  is the number of communication configurations with  $s$  active sources and  $t$  engaged TSVs, while  $(N_r - 1)^s$  is the number of possible combinations of destinations for an arrangement of  $s$  active sources.

The first point of contrast between  $N(s, t)$  and  $(N_r - 1)^s$  is that the former spans *all* possible arrangements of  $s$  active sources, while the latter points to a *specific* arrangement of  $s$  active sources. This contrast has to be removed in order to find an intuitive interpretation for the ratio between  $N(s, t)$  and  $(N_r - 1)^s$ . Since the number of possible arrangements of  $s$  active sources is equal to  $\binom{N_r}{n}$ , the expression  $\binom{N_r}{n}(N_r - 1)^s$  can be interpreted as the number of possible combinations of destinations for *all* possible arrangements of  $s$  active sources rather than for a *specific* arrangement of  $s$  active sources. Therefore, the ratio  $N(s, t)/(N_r - 1)^s$  in Equation 3.7 can be rewritten as:

$$\frac{N(s, t)}{(N_r - 1)^s} = \frac{N(s, t)}{\binom{N_r}{s}(N_r - 1)^s} \binom{N_r}{s} \quad (3.8)$$

Given that all communication configurations with  $s$  active sources are equally likely, the ratio:

$$\frac{N(s, t)}{\binom{N_r}{s}(N_r - 1)^s} \quad (3.9)$$

can be interpreted as “the probability of having  $t$  engaged TSVs, provided that  $s$  sources are active”. Denoting this probability by  $P_{T|S}(t|s)$ , Equation 3.7 can be written as:

$$P_{f,\text{system}} = \sum_{s=0}^{N_r} \sum_{t=0}^{N_t} \left[ \binom{N_r}{s} P_{T|S}(t|s) \times \alpha^s (1 - \alpha)^{N_r - s} [1 - (1 - P_{f,\text{TSV}})^t] \right] \quad (3.10)$$

The probabilistic interpretation of Equation 3.9 suggests that  $P_{T|S}(t|s)$  can be approximated by Monte Carlo simulation. Specifically, a sufficiently large number of random communication configurations with  $s$  active sources is generated. Then, the fraction of

outcomes with the number of engaged TSVs equal to  $t$  is calculated. The results can be used to approximate  $P_{T|S}(t|s)$ . The larger the sequence of generated configuration, the more accurate is the approximation. This method can be implemented by a simple algorithm. In the first step, a random arrangement of  $s$  sources is generated. In the second step, the corresponding destination locations are also generated randomly. Finally, the set of TSVs required to support the data links between these sources and destinations is calculated and the number of engaged TSVs is recorded. These steps are repeated for a sufficiently large number of times and the record is updated every time. Using the record of number of engaged TSVs, one can easily estimate  $P_{T|S}(t|s)$  for a fixed  $s$  and all  $t$ 's. Algorithm 3.1 shows the pseudo-code for estimation of  $P_{T|S}$ . The notation  $M(:,j)$  for a matrix  $M$  denotes the  $j^{th}$  column of matrix  $M$ . The notation  $\mathbf{0}_{M \times N}$  represents an  $M \times N$  matrix with all elements set to 0.

The error function in line 18 of Algorithm 3.1 calculates a “distance” between two consecutive estimates of  $P_{T|S}(t|s)$  for a fixed  $s$ . Specifically, the distance is:

$$\text{err}(P_{T|S}(:,n_s), \hat{P}_{T|S}(:,n_s)) = \frac{\sum_{i=1}^{N_r} |(P_{T|S}(i, n_s) - \hat{P}_{T|S}(i, n_s))|}{\sum_{i=1}^{N_r} \hat{P}_{T|S}(i, n_s)} \quad (3.11)$$

which is a norm-1 relative error.

If this error is less than a threshold  $c_{th}$ , the simulation stops and the last estimate of  $P_{T|S}(t|s)$  is used. One may argue that two consecutive estimates of  $P_{T|S}(t|s)$  may turn out to be equal prematurely, i.e. before the estimate of  $P_{T|S}(t|s)$  has sufficient accuracy. To reduce the chance of such an occurrence, the condition on error is calculated and checked every hundred iterations rather than every iteration.

---

**Algorithm 3.1.** Estimation of  $P_{T|S}(t|s)$ 

---

```
1: Input: Convergence threshold  $cth$ ,  $N_t$ ,  $N_s$ , routing algorithm and network geometry.
2: Output: Estimate of  $P_{T|S}(t|s)$ .
3:  $TSVcount = \mathbf{0}_{(N_t+1) \times (N_s+1)}$ 
4:  $P_{T|S}(t|s) = \mathbf{0}_{(N_t+1) \times (N_s+1)}$ 
5: for  $n_s := 0$  to  $N_s$  do
6:   for  $n_t := 0$  to  $N_t$  do
7:      $Counter = 0$ 
8:      $Convergence = True$ 
9:     while  $Convergence = True$  do
10:       $Convergence = False$ 
11:       $PrevTSVcount = TSVcount$ 
12:       $\hat{P}_{T|S} = P_{T|S}$ 
13:       $++ Counter$ 
14:      Choose  $n_s$  random sources
15:      Choose  $n_s$  random destinations
16:      Calculate number of active TSVs based on routing algorithm
17:       $n_t =$  Number of active TSVs
18:       $++ TSVcount(n_t, n_s)$ 
19:      if  $Counter \neq 0$  then
20:         $P_{T|S}(:, n_s) = TSVcount(:, n_s) / Counter$ 
21:        if  $err(P_{T|S}(:, n_s), \hat{P}_{T|S}(:, n_s)) \leq cth$  then
22:           $Convergence = True$ 
23:        end if
24:      end if
25:    end while
26:  end for
27: end for
```

---

A small value of  $c_{th}$  ensures higher accuracy in estimation of  $P_{T|S}(t|s)$  while imposing a longer simulation time. Consequently, a proper value of  $c_{th}$  should be selected to balance the trade-off between accuracy and simulation time. To find a proper value for  $c_{th}$ , the variation of error function in Equation 3.11 versus iterations is examined. The error function is a function of  $n_s$  and routing algorithm. Therefore, examination of error function versus number of iterations requires specifying these two factors.

Figure 3.1 illustrates a sample logarithmic diagram of error function versus iteration count for an  $8 \times 8 \times 8$  router network, supported by an  $8 \times 8 \times 7$  TSV network, with  $n_s = N_r/4$  and dimension order routing. Interestingly, the variation is linear in logarithmic scale. Consequently, one may estimate the number of required itera-

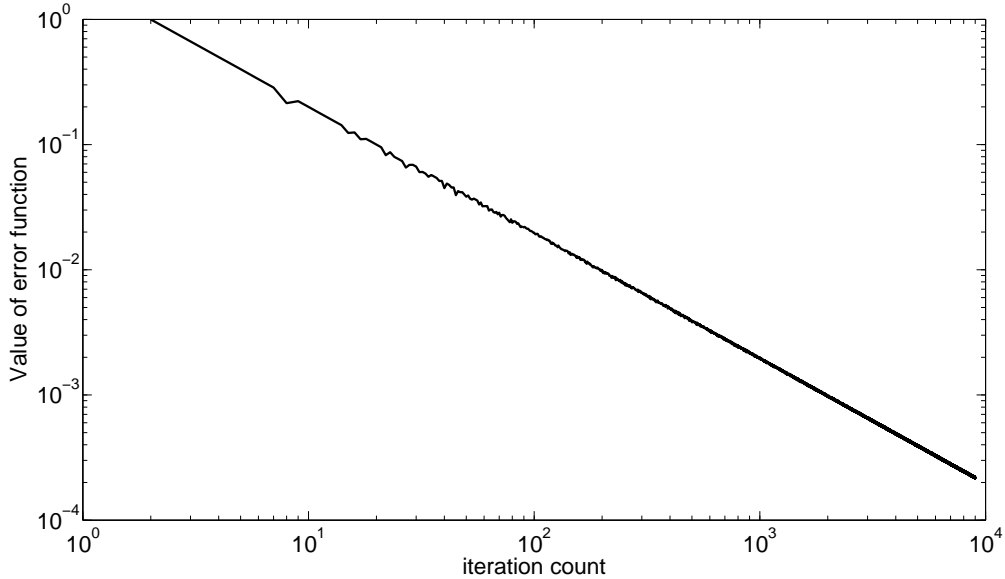
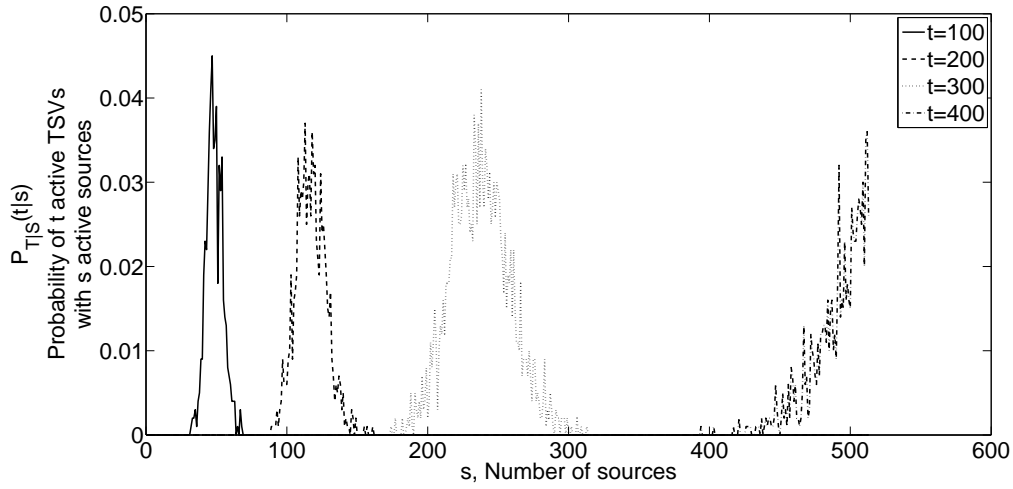
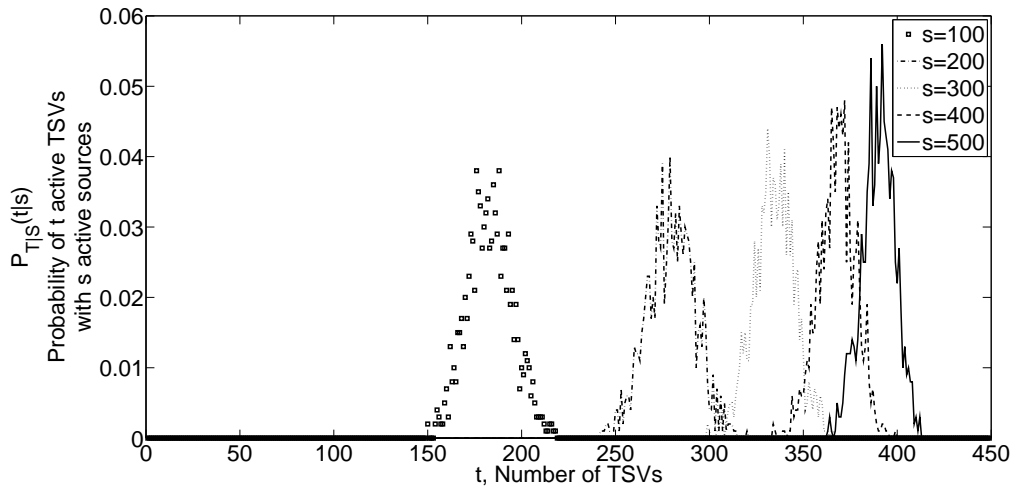


Figure 3.1: A sample of error function vs number of iterations for an  $8 \times 8 \times 8$  router network supported by and  $8 \times 8 \times 7$  TSV network,  $n_s = N_r/4$ , with dimension order routing.

tions for a given accuracy through a linear regression of Figure 3.1. The simulations show the same linear behavior for different IP/TSV geometrical placements, different routing algorithms, and other values of  $N_r$  and  $n_s$ . It is noteworthy that line 13 of Algorithm 3.1, which is responsible for calculating the set of TSVs required to support the data links between generated sources and destinations, depends on how the architecture handles data link requests of sources. For example, different architectures may have different scheduling or buffering strategies when several sources need to use a common set of TSVs. To simplify the simulation, it is assumed that the aggregate set of engaged TSVs is the union of the sets of TSVs engaged to support the data link between individual source-destination pairs. Furthermore, routers are assumed not to use any buffering mechanism. However, as long as the aggregate output traffic of a router can be modeled by the concept of injection rate (technically, a two state



(a)  $P_{T|S}(t|s)$  as a function of number of sources, with number of TSVs as plot parameter, and for an  $8 \times 8 \times 8$  network. Created with MATLAB<sup>®</sup>.



(b)  $P_{T|S}(t|s)$  as a function of number of TSVs, with number of sources as plot parameter, and for an  $8 \times 8 \times 8$  network. Created with MATLAB<sup>®</sup>.

Figure 3.2: Variation of  $P_{T|S}(t|s)$  with number of active sources and TSVs

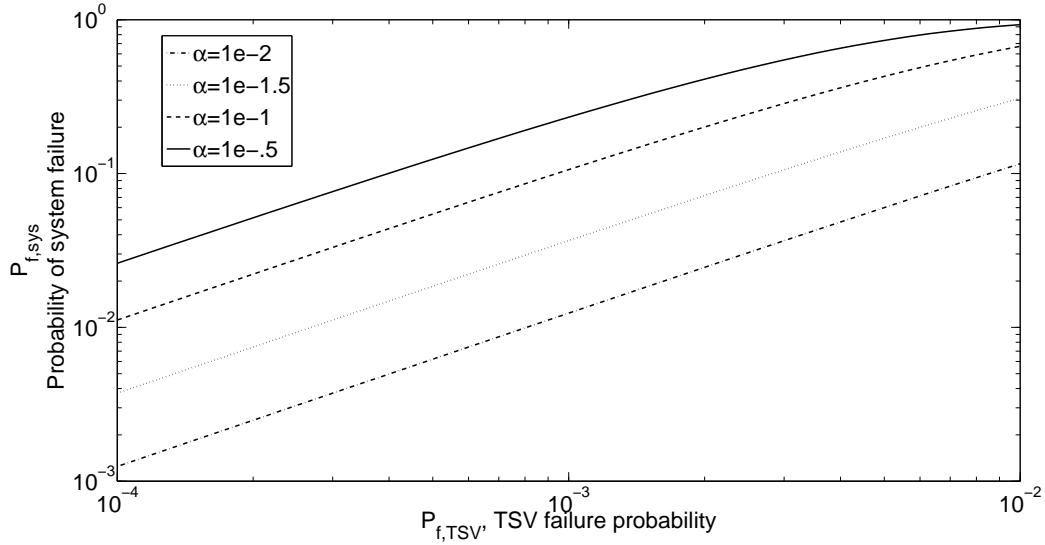
Markov chain), the same concept and probabilistic interpretation can be used with other architectures deploying different routing and buffering mechanisms.



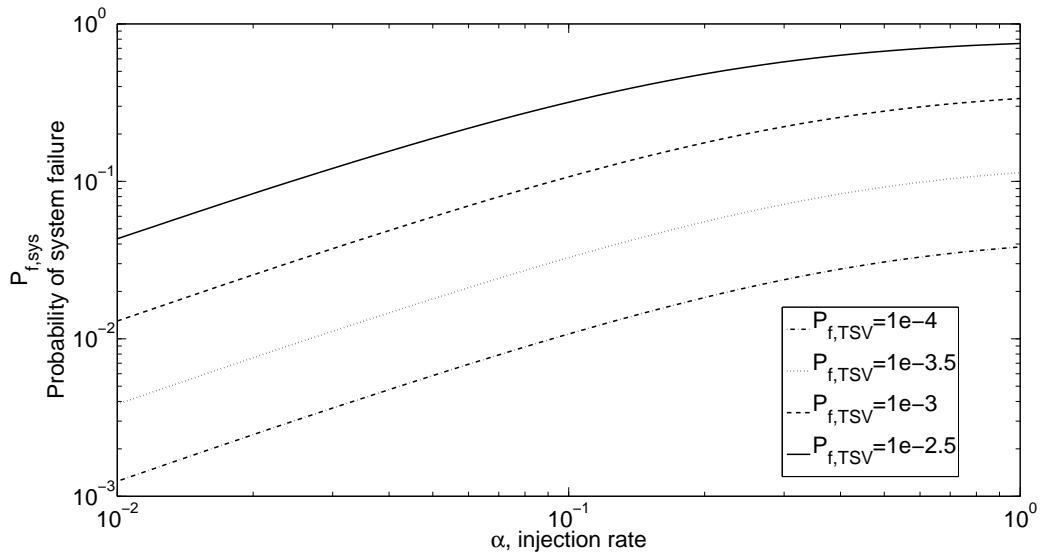
Figures 3.2a and 3.2b illustrate slices of the 2D function  $P_{T|S}(t|s)$  for an  $8 \times 8 \times 8$  network of routers supported by an  $8 \times 8 \times 7$  network of TSVs. Figure 3.2a is parametrized by the number of active TSVs, while Figure 3.2b is parametrized by the number of sources. The noise-like fluctuations are due to the limited number of Monte Carlo runs used to estimate the probability and can be reduced by lowering the discussed convergence threshold  $c_{th}$ . It is observed that the probability distribution for a fixed number of sources is very close to a Gaussian whose mean and variance depends on the number of sources. On the other hand, the distribution for a fixed number of active TSVs is more of a truncated Gaussian.

Algorithm 3.1 repeats until the error between two consecutive estimates of  $P_{T|S}(t|s_0)$  falls below  $10^{-3}$ . Furthermore, dimension-order routing is used: if a router at location  $(x_1, y_1, z_1)$  wants to communicate with a router at location  $(x_2, y_2, z_2)$ , the set of locations of engaged TSVs is  $\{(x_2, y_2, z)|z \in \{\min(z_1, z_2), \dots, \max(z_1, z_2) - 1\}\}$ . Finally,  $P_{T|S}(t|s)$  can be used in Equation 3.10 to find  $P_{f,system}$  for different values of injection rate and probability of TSV failure. As depicted in Figure 3.3 and in accordance with intuition, it is observed that the higher the injection rate and TSV failure probability, the higher is the probability of system failure. Interestingly, the probability of system failure is observed to follow an almost linear dependence on injection rate and TSV failure probability in the logarithmic scale, especially at lower values of injection rate and TSV failure probability.

It is important to note that the computational attractiveness of this approach is twofold. First, an estimate of  $P_{T|S}(t|s)$  is readily obtained through a Monte Carlo simulation. Second, integrating the effect of injection rate and TSV failure probability into  $P_{f,system}$  is facilitated by an *analytic* expression given by Equation 3.10 rather than



(a) Probability of system failure as a function of TSV failure probability ( $P_{f,TSV}$ ). Created with MATLAB<sup>®</sup>.



(b) Probability of system failure as a function of injection rate ( $\alpha$ ). Created with MATLAB<sup>®</sup>.

Figure 3.3: Variation of system failure probability with injection rate and TSV failure probability

requiring a separate time-consuming simulation for every choice of  $\alpha$  and  $P_{f,TSV}$ .

### 3.3 Summary

The challenges of implementing many-core systems on a 3D-NoC using TSV technology as vertical interconnect were presented. After modeling TSV characteristics as a time-invariant failure probability, a reliability criterion for TSV-based NoC was defined as the probability of having at least one faulty TSV in a given time slot. The relationship between NoC reliability and TSV failure was quantified, and the resulting equation was reduced to a tractable form with manageable computational complexity. The final equation relating NoC reliability criterion and TSV failure includes a non-analytical probabilistic term which can be efficiently approximated by the Monte Carlo simulation for different architectures. Importantly, the simulation only depends on network geometry and routing algorithm, and the effect of injection rate and TSV failure is decoupled from the simulation. Therefore, the result of simulation can be used to calculate the reliability criterion for a wide range of injection rate and TSV failure rate values. This is in contrast with the traditional approach in which a separate simulation run is required for each value of injection rate and TSV failure rate. Finally, the reliability criterion of a simple  $8 \times 8 \times 8$  NoC supported by a  $8 \times 8 \times 7$  network of TSVs was calculated to demonstrate the numerical evaluation of the proposed method.

# Chapter 4

## System-level TSV Coupling Fault Model

### 4.1 Introduction

The reliability of TSVs is a underresearched topic in the context of designing 3D stacked chip systems. TTCC, the main focus of this chapter, is one of the major challenges in designing 3D stacked IC designs. TTCC is a result of rising parasitic signal, which may cause two major issues. First, it increases the path delay due to the Miller effect. This effect slows down transitions on signal-switching, if neighbor TSVs perform opposite transitions. Second, the coupling noise may result in signal distortion by generating large glitches on a static signal when TSV's aggressor neighbors transition [62].

Reliable 3D ICs have been proposed by many research groups [3, 19], but these

approaches are all evaluated by assuming uniform random faults distribution in time of occurrence and spatial location. An analytical model for the coupling capacitance between pairs of TSVs is presented in [94]. The impact of TSVs on SI in 3D ICs has been investigated in several articles [94, 109]. However, evaluation of TTCC effects by injecting random fault distribution is 26%-99% inaccurate in capturing time and locations of induced faults, as discussed in Section 4.4.2. Therefore, system-level designers cannot accurately report the fault-prone components of TSV-based 3D ICs and protect them accordingly. In another similar research, a system-level process variation model has been proposed for 2D NoC simulators [2], but it does not support TSV coupling effect in 3D ICs.

In this chapter, I present an accurate system-level fault model to quantify the system-level impact of TSV coupling-induced faults at runtime, pinpoint fault-prone TSV link connections that should be protected. It also accurately evaluates alternative resilient TSV-based 3D IC designs by considering circuit-level TSV characteristics and thermal impact for a given data application. Having analyzed and recorded the TSV coupling effect at circuit-level, these effects are applied to the TSVs in system-level simulations at runtime through precise monitoring and calculation. The proposed fault model is potentially useful for evaluating reliability of 3D many-core applications in which TSV coupling may lead to failure. 3D memory and on-chip network of routers are some examples of these applications. In 3D memory applications [45], data or address buses may get corrupted due to TSV coupling. In on-chip network routers, data flit corruption may occur due to TSV coupling which needs an Error-Correction Code (ECC) to tolerate corruption.

In this chapter, I elaborate the TTCC effects on timing requirement of a circuit

in order to present circuit-level fault library (discussed in Section 4.2). The proposed fault library is invoked at system-level with the goal of TSV coupling effect mapping automation. Then, I provide a system-level TTCC fault modeling tool for TSV-based 3D ICs (presented in Section 4.3). This model is capable of being plugged into any data-oriented simulator to detect and inject TSV coupling faults at runtime, to evaluate resilient approaches. At the end, I present a case study, characterizing faults at runtime in a  $4 \times 4 \times 4$  3D-NoC architecture (explained in Section 4.4).

## 4.2 TTCC Elaboration

As stated earlier in Chapter 2, the term TSV coupling refers to capacitive and inductive couplings among adjacent TSVs. This chapter targets the capacitive coupling effect which is more critical in lower range of operational frequencies (less than 5GHz) [7]. This is because, there is no intent of frequency escalation due to extra power consumption, fabrication issues, and unexpected heat generation in higher frequencies [44]. In this section: first, our circuit-level model of TTCC is discussed, although any other models can be replaced; second, a TTCC classification is presented; finally, the effect of TTCC on timing requirement of circuit is illustrated using the traffic traces of PARSEC applications.

### 4.2.1 TTCC Circuit-level Modeling

A framework consisting of multiple TSVs at circuit-level using Synopsys HSPICE is implemented in an experiment to study the sources of TTCC effect. Developing TSV

$$\begin{aligned}
\mathbf{Data}_{i-1} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} & \mathbf{Data}_i &= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \\
\mathbf{Current\ flow} &= \begin{bmatrix} \textcircled{\circ} & \textcircled{\odot} & \textcircled{\circ} \\ \textcircled{\odot} & \textcircled{\otimes} & \textcircled{\circ} \\ \textcircled{\circ} & \textcircled{\otimes} & \textcircled{\circ} \end{bmatrix} \\
\mathbf{TTCC} &= \begin{bmatrix} 2 & 4 & 1 \\ 4 & \textcircled{5} & 1 \\ 2 & 2 & 1 \end{bmatrix}
\end{aligned}$$

Figure 4.1: Current and TTCC matrices in a  $3 \times 3$  mesh of TSVs

simulation framework allows to extract the realistic accurate TSV coupling effect for different parameters. The coupled TSV structure is modeled as a lumped RLC circuit including multiple signal TSVs [62, 55, 83], as shown in Figure 4.2. The circuit is composed of a series TSV resistance  $R_{TSV}$  and inductance  $L_{TSV}$ , parallel silicon substrate resistance  $R_{si}$  and capacitance  $C_{si}$ , and silicon dioxide capacitance  $C_{ox}$  around TSV. The value of the circuit elements is modeled using analytical equations based on the dimensions of the structure, such as oxide thickness, silicon substrate height, TSV radius, and TSV pitch and by material properties like dielectric constant and resistivity. The thermal impact is also considered in the TSV model using equations in [61].

In this framework, TSV is connected to the output of an inverter (driver) on one side and to the input of another inverter on the other side (load). These inverters are needed to record delay and its dependency on parasitic capacitive coupling. Two flip-flops, one before the driver inverter and one after the load inverter are inserted

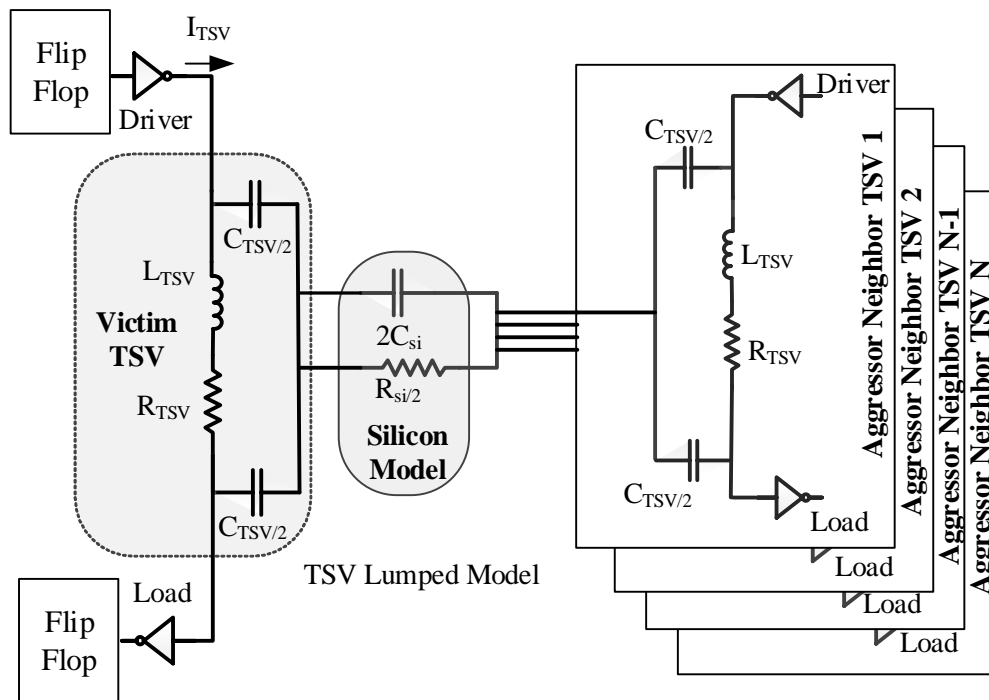


Figure 4.2: Measure the induced capacitive coupling on each TSV based on physical parameters

to capture the parasitic capacitive coupling effects on timing requirements of circuit. The input data pattern is compared with output data pattern to catch the parasitic capacitive coupling effects. Predictive Technology Model (PTM) [88] FinFET transistor models are employed to implement inverters and flip-flops. Then, a comprehensive set of simulations is performed on the developed TSV framework. The impact of operational frequency, temperature, technology, TSV radius, and TSV oxide thickness are investigated and shown in Figure 4.3. SPICE model of TSVs are employed to examine the TTCC effect among a victim and its aggressor TSVs. In this chapter, path delay is defined as the time data needs to propagate from the input of the first flip-flop to the output of the load inverter after the the rising edge of the clock. Nominal Path Delay (NPD) is the path delay when there is no TSV parasitic capacitive coupling.



Actual Path Delay (APD) may be longer than NPD due to the coupling from aggressors. Assuming the system clock is adjusted for a critical path of NPD, circuit timing requirement is violated when the APD exceeds the NPD. Timing Violation (TV) is defined as any additional delay over NPD (introduced by parasitic capacitive coupling) normalized by clock period, as presented in Equation 4.1:

$$TV = \frac{APD - NPD}{T_{clk}} = f_{clk}(APD - NPD) \quad (4.1)$$

Running some simulation on the developed circuit model with different input data patterns, different TVs are reported; concluding that TTCC values are data dependent. Also, as explained earlier in Chapter 2, the TTCC parasitic value is a function of the electrical potential differences between a TSV and its neighbors [58]. Moreover, the electrical potential differences depends on the current flow of a given TSV and its neighbors. On the other hand, the current flow direction of a TSV is data-dependent, based on charging and discharging of the intermediate capacitor between transistors pairs located in different stacked layers. Furthermore, it is observed that different data bit patterns driving TSVs, result in similar TV on a same TSV model. This explanation confirms our previous observation in which TTCC values are data dependent and hence predictable. In other words, the TTCC effect on circuit timing is predictable by monitoring the data bit patterns fed into the TSVs, which is discussed in the following Subsection.

In addition, these capacitive coupling values disrupt timing requirement of 3D-IC based on the operational and TSV physical parameters. The effect of TTCC is characterized for each of parasitic capacitive coupling types presented in Table 2.3 in our circuit-level model. This characterization for a range of operational frequency and

different TSV parameters is elaborated in Figure 4.3.

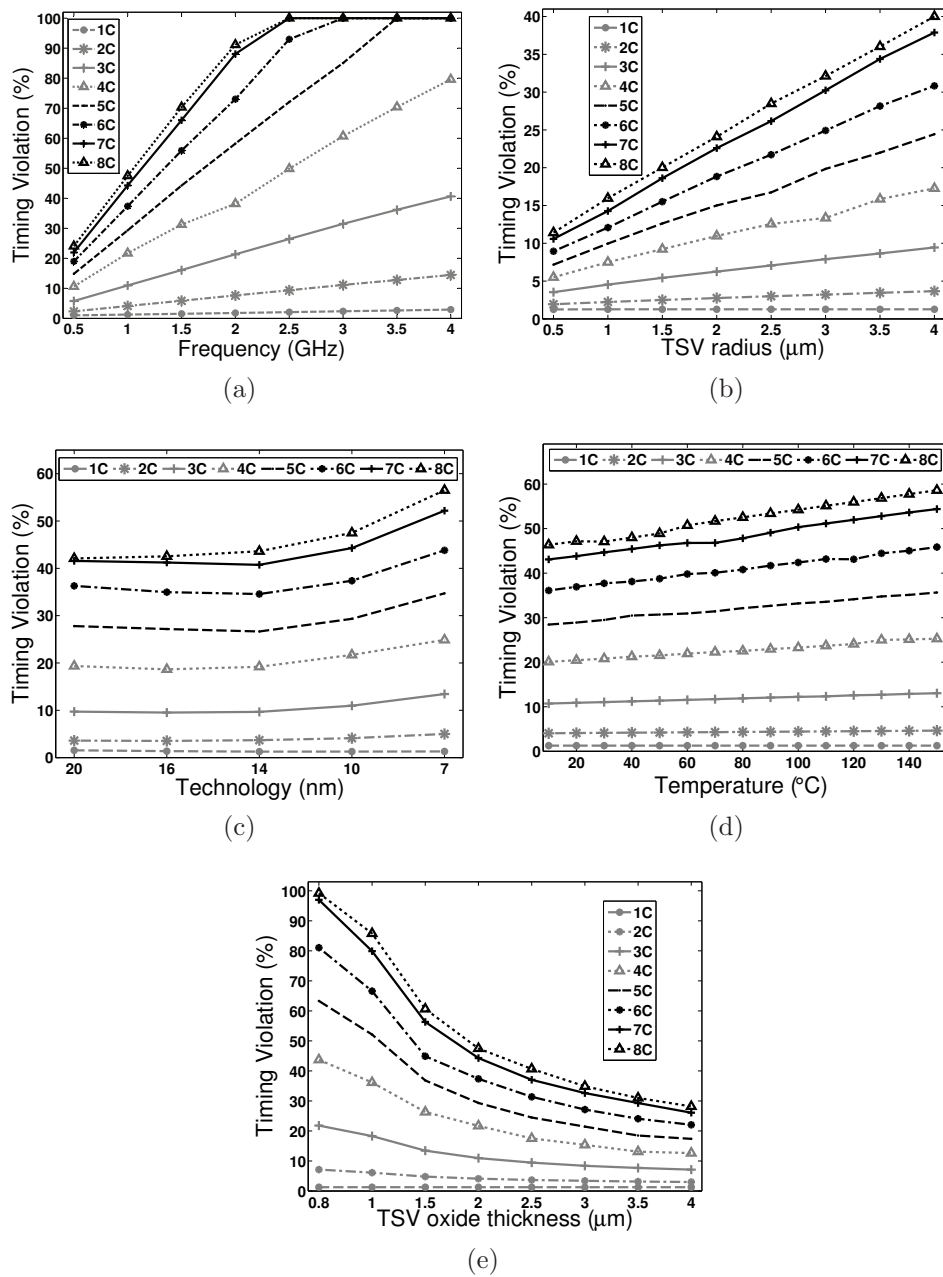


Figure 4.3: Characterizing TSV coupling against various parameters

In Figure 4.3a, increasing clock frequency does not have tangible effect on the TTCC severity, but TV is increasing linearly for larger operational frequencies. This

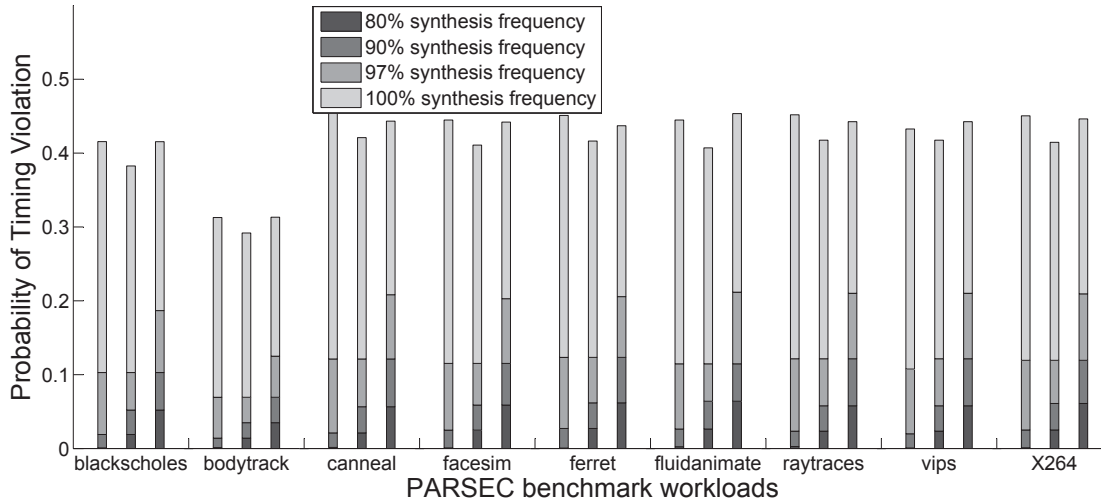


Figure 4.4: Probability of TV in PARSEC benchmark workloads under different conditions

is because timing requirement gets tight in higher frequencies. For TSVs with larger radius with same pitch value,  $C_{si}$  increases and therefore more capacitive coupling is observed, as shown in Figure 4.3b. As the technology advances, the loading voltage of the flip-flop over the TSV decreases, resulting in larger coupled voltage on the TSV (shown in Figure 4.3c). The permittivity of the silicon rises as a weak linear function of temperature [61], which results in an increase in  $C_{si}$  and increases coupling and TV, as depicted in Figure 4.3d. Finally, as shown in Figure 4.3e, thicker oxide provides better isolation and reduces the value of  $C_{ox}$ , resulting in less capacitive coupling and consequently less TV.

## 4.2.2 TTCC Effect on TV

Having analyzed and modeled the TTCC, the TTCC effect is evaluated with realistic data to emphasize on the necessity of TTCC consideration. The realistic data is

Table 4.1: Simulation configuration settings

Configuration	A	B	C
radius $\mu m$	3	3	2
length $nm$	15	20	25
pitch $\mu m$	9	9	6
$T_{ox}$ $\mu m$	3	2.5	1
Technology $nm$	20	16	10
Max Freq. $GHz$	1	1	1
Temperature $^{\circ}C$	50	75	100

the memory traces which are collected via Pin tool via running PARSEC benchmark workloads on a real system. These data are fed into the circuit-level model with 64 TSVs for various configurations and the TV for each TSV, if occurred, is recorded. A configuration is a set of TSV physical parameter values including TSV radius, length, pitch, oxide thickness, and process technology (transistor), operating frequency, and temperature. The configuration values are selected in a way to cover different TTCC effects. Also, various timing specification (setup/hold-time) is considered on the receiver side to create a real scenario. Figure 4.4 shows the probability of TV for different workloads with different configurations at different synthesis frequencies. Bars in the figure from left to right refers to configurations A, B, and C, respectively. An average of 40% TV at 100% synthesis frequency confirms the importance of TTCC analysis in TSV-based 3D architectures.

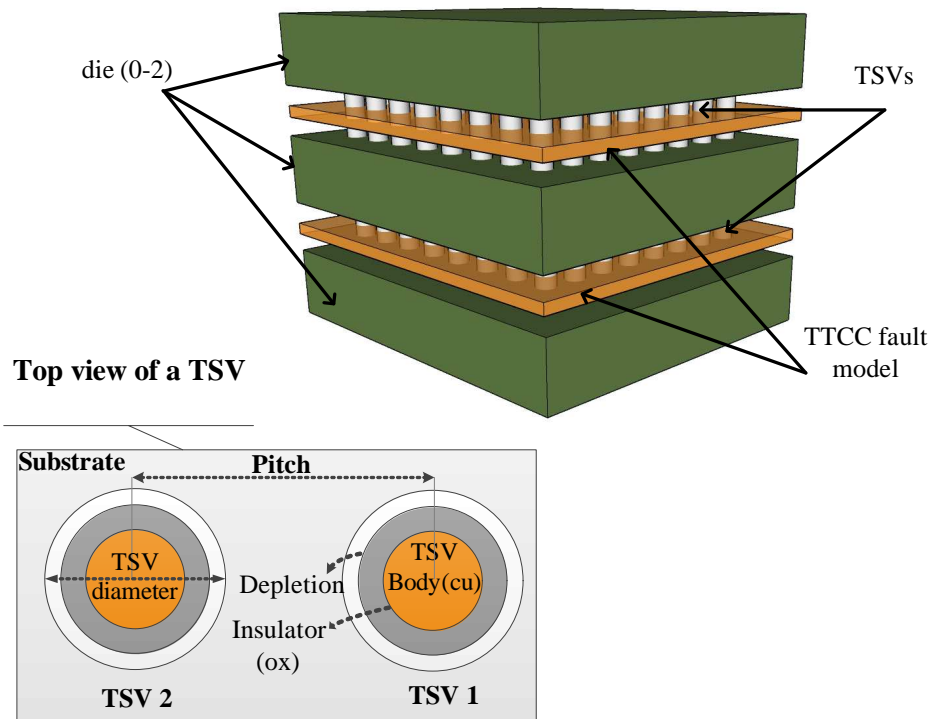


Figure 4.5: 3D IC structure with the proposed capacitive coupling fault model

### 4.3 TSV Coupling Fault Model

Circuit-level simulation takes longer than system-level one with multiple orders of magnitude. In this chapter, I propose a fault model which is capable of modeling accurate TTCC faults. This model employs the effects of circuit-level model of TTCC in system-level platform to save simulation time while it guarantees the accuracy. This operation is performed at runtime by monitoring the data bits feeding TSVs in order to identify the location and time of potential faulty TSVs. Then, the candidate faulty TSVs are triggered with an appropriate consequence, as circuit-level effect of TTCC fault.

Figure 4.5 and Figure 4.6 show the 3D-IC framework and the proposed TSV capacitive coupling fault model in detail. The devised fault model is supposed to be employed as an intermediate component among TSVs connecting ICs in different dies, as shown in Figure 4.5. This fault model does not affect the functionality of 3D-IC; it only decides the time and location of fault activation through the TSVs based on data input patterns and provided fault library by the circuit model. Figure 4.6 depicts the functionality of this fault model in detail. The input parameters of the circuit-level model are TSV arrangement configuration (number of rows and columns) connected to 3D-IC, operating frequency, process technology, silicon oxide thickness, TSV-to-TSV pitch, TSV length, and TSV diameter. The effects of these operational and TSV physical parameters on timing requirement of a fixed 3D design have been discussed in Subsection 4.2.1. The output of this model is a table, which provides the corresponding parasitic capacitive coupling values causing timing violation for each configuration. This output is used as a fault library in the system-level simulation, in which the parasitic capacitive coupling values are extracted by comparing the transmitted ( $Data_{i-1}$ ) and ready to transmit ( $Data_i$ ) data bits through TSVs. With this configuration, the TTCC fault model decides intelligently and accurately when and where a TTCC fault should be activated.

**This methodology is executed in the following five steps:**

- **Step.0 Configuration and Setup:**

Prior to instantiating and utilizing the devised TSV coupling fault model, first it needs to be configured and setup. In this phase, input parameters of the model such as TSV length, TSV diameter, TSV pitch, oxide thickness of TSVs, process technology, frequency, and temperature are specified. Frequency and

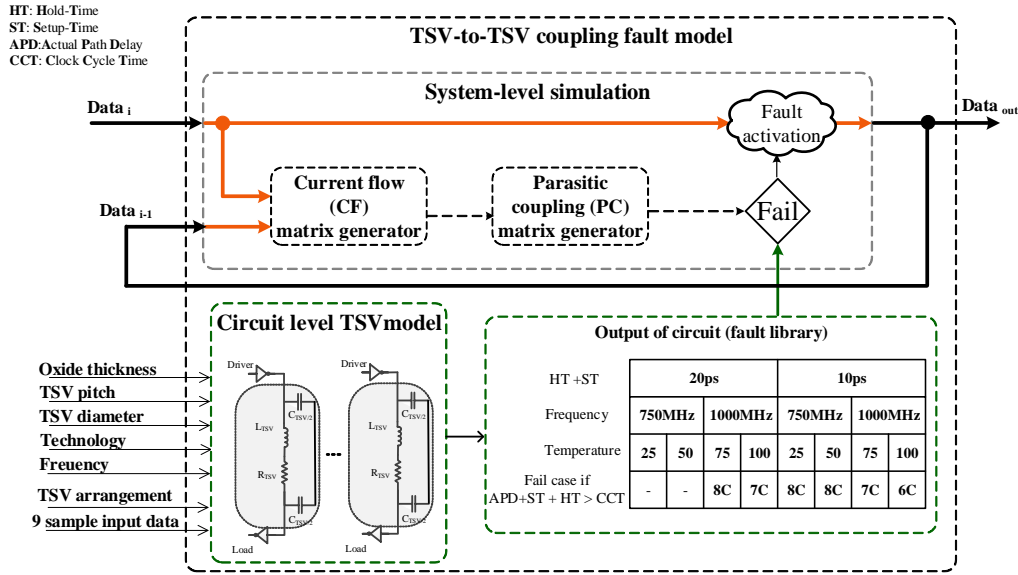


Figure 4.6: Fault model usage demonstration in 3D-IC

temperature parameters are defined as a range with a specific granularity to support dynamic changes at run time. Also, all possible data inputs resulting in 9 parasitic capacitive coupling values, from 0C to 8C, are the other input of the circuit-level model and the outcome is recorded in a table as the fault library. In other words, this table includes timing delay for each case and configuration. Figure 4.6 shows a sample fault library table. Having characterized the fault library, at runtime phase multiple steps are taken, in order to accurately map the TSV coupling effect at system-level which are categorized as Step-1 to Step-4:

- **Step.1 Capturing transferred data:**

In this step, the data bits transferring through TSV links (Up/Down port) are captured as the input of fault model at run-time. These captured data bits are adjusted if the fault activation condition is satisfied.

- **Step.2 Data analysis to determine the TSV current flows:**

At this step, the current flow direction of all TSVs are identified. The previous ( $Data_{i-1}$ ) and current ( $Data_i$ ) data bit value of a TSV's driver are profiled and compared in order to recognize the current flow of each TSV. This process is done with the same approach, as discussed in Chapter 5. The output of this stage is stored as Current Flow (CF) matrix in a way that each matrix element corresponds to a TSV with the same row and column index.

- **Step.3 Determine the induced capacitive coupling case for each TSV:**

Looking at the current flow direction of each TSV and its adjacent neighbors, in this step, an appropriate capacitive case is assigned per TSV. Now, the fault library generated at pre-runtime is used to look up the timing delay associated with the corresponding case and recorded as Parasitic Coupling (PC) matrix.

- **Step.4 Map circuit-level faults to system level delay fault/failure for each TSV:**

Considering the setup-time parameter of the destination receiver flip-flop, at this point, using PC matrix, the decision for faulty TSVs is made and an appropriate fault is applied to the detected faulty TSVs. If the reported timing delay associated with the capacitive case is more than the hold-time of receiver then the data bit is assumed faulty. The fault type, depending on the specified delay tolerance for the circuit, can be either a "*delay fault*" or a "*glitch fault*." However, if the specified flip-flop hold-time is long enough, then the "*delay fault*" will not have any impact and it is ignored in the model. Now, the data is ready to be forwarded towards destination. Also, a copy of last transmitted data is maintained for subsequent current flow direction determination.



Table 4.2: System configuration parameters

Parameter	Value
Topology	3D fully connected Mesh
Network Size	(4x4x4) 64 Routers
Flit Size	32 bits
Buffer Depth	8, 16-bit entries per port
Switching Scheme	Wormhole
Routing Algorithm	Dimension Ordered Routing
Simulator	THENoC [102]

## 4.4 Case Study: Diagnosing TSV Coupling at Runtime in 3D-NoC

A case study is presented in this section to show the usage of the proposed TTCC fault model at system-level; first, the simulation infrastructure details is discussed; second, the accuracy of the fault model is compared; and finally, the number of TTCC faults due to timing violation in each data transaction under PARSEC benchmarks is reported.

### 4.4.1 System Configuration

A cycle-accurate simulation, on the 64-node 3D mesh NoC (the simulation parameters are shown in Table 4.2) is performed. The proposed TTCC fault model is implemented at the system-level in C. In the developed 3D-NoC, TSV bundles between each router in different layer are connected through the proposed coupling fault model.

This fault model locates and triggers the parasitic capacitive coupling for the

specific configuration of TSVs with the given physical parameters, as elaborated in Section 4.3. Prior to running the simulation, each configuration is run in HSPICE and the result is passed as a static configuration library to the C model. Using the information provided by HSPICE (fail cases) and the CD matrix, the fault model detects the coupling fault and triggers it accordingly on-the-fly. The fault model operation is detailed in Section 4.3 and is illustrated in Figure 4.6.

#### 4.4.2 Fault model accuracy

In order to demonstrate the accuracy of our fault model, a comparison is made with most commonly used crosstalk fault model using popular uniform random fault distribution [2] for system-level simulation of reliability study of NoC. The comparison shows that the accuracy of conventional fault model for TSVs is substantially inaccurate.

First, a 3D-NoC simulation with the TSV coupling fault model using random traffic is performed and the number of fault occurrence for a TSV bundle is calculated. Then, using conventional fault model, the 3D-NoC simulation is run for 10000 times and each time faults are randomly injected across the TSV bundles to measure the inaccuracy that is introduced by such a distribution. These simulations are repeated considering different configurations in which failing case might be 8C, 7C, 6C, 5C, 4C, or 3C. Considering that each capacitive case has a different probability of occurrence, as depicted in Table 2.3, these probabilities are considered in random fault injection simulations. For example, if 6C parasitic coupling results in failure based on the circuit-level modeling in a given configuration, the failure probability of a TSV in random simulation will be equal to the occurrence probability of 8C, 7C, or 6C which is  $10/81$ .

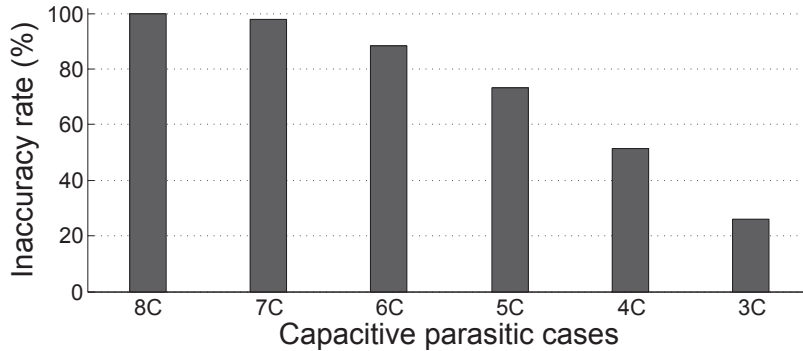


Figure 4.7: Inaccuracy introduced by random fault models

For a fair comparison, the occurrence probability of capacitive coupling higher than 4C and 6C is extracted for TSVs on the corner and boundaries in a TSV bundle.

Figure 4.7 shows the inaccuracy introduced by random fault model distribution for different TSV configuration (leading to different failing cases). 0% inaccuracy corresponds to the distribution matching the model’s probability of occurrence, while 100% inaccuracy implies that the random distribution does not predict that the corresponding fault type will occur. It is observed that randomly distributed faults across the  $8 \times 8$  TSV bundle introduces almost 99% inaccuracy for a given TSV specification leading to 8C capacitive coupling case. Because of more fault occurrence for in lower fail cases and hence more matching probability, the percentage of inaccuracy decreases.

### 4.4.3 TSV Coupling Fault Characterization

In order to demonstrate the effect of network traffic on TSV coupling fault, the PARSEC benchmark traces are collected with GEM5 full system simulator, and then injected them into the  $4 \times 4 \times 4$  3D NoC simulator (THENoC). Figure 4.8 shows the

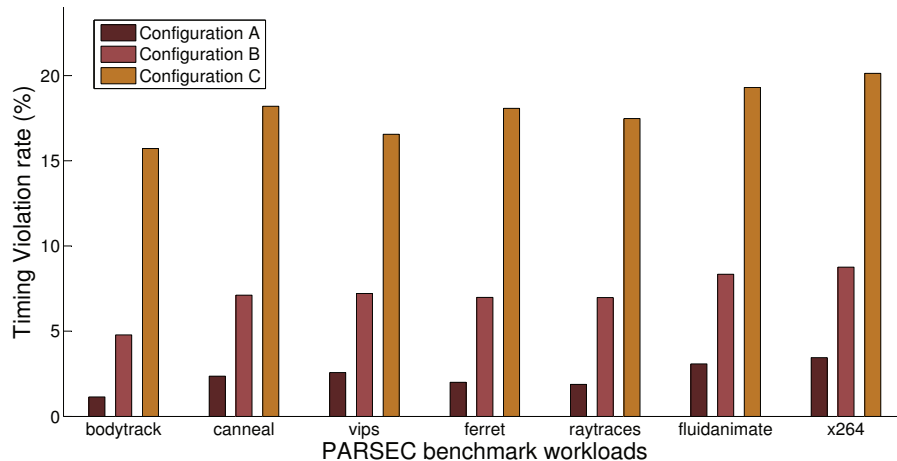


Figure 4.8: Fault model usage demonstration in 3D-NoC running PARSEC benchmark ratio of timing violation occurrence over total number of data transactions in vertical direction for three different configuration (described in Figure 4.4) at 90% synthesis frequency. For *Configuration C*, an average of 18% timing violation is reported because of running at higher temperature and low TSV oxide thickness which exasperate the TTCC. As the Configuration parameter values get relaxed, the timing violation due to TTCC is also decreases accordingly.

TTCC fault density map for the  $4 \times 4 \times 4$  simulated network with *canneal* workload traffic is depicted in Figure 4.9. Each subfigure represents  $4 \times 4$  NoC routers of a specific layer sending data to their lower/upper layer routers (*Down port/Up port*). The values are normalized to the maximum number of timing violation in entire 3D-NoC. The layer 0 down port and layer 3 up port are not shown since they do not exist. With this map, designers can have better knowledge of placing their resiliency methods for an specific application. It can be seen that the data transactions from *layer 2* going down to *layer 1* cause a lot of capacitive coupling faults.

These information can help the 3D-IC designers to easily and accurately assess

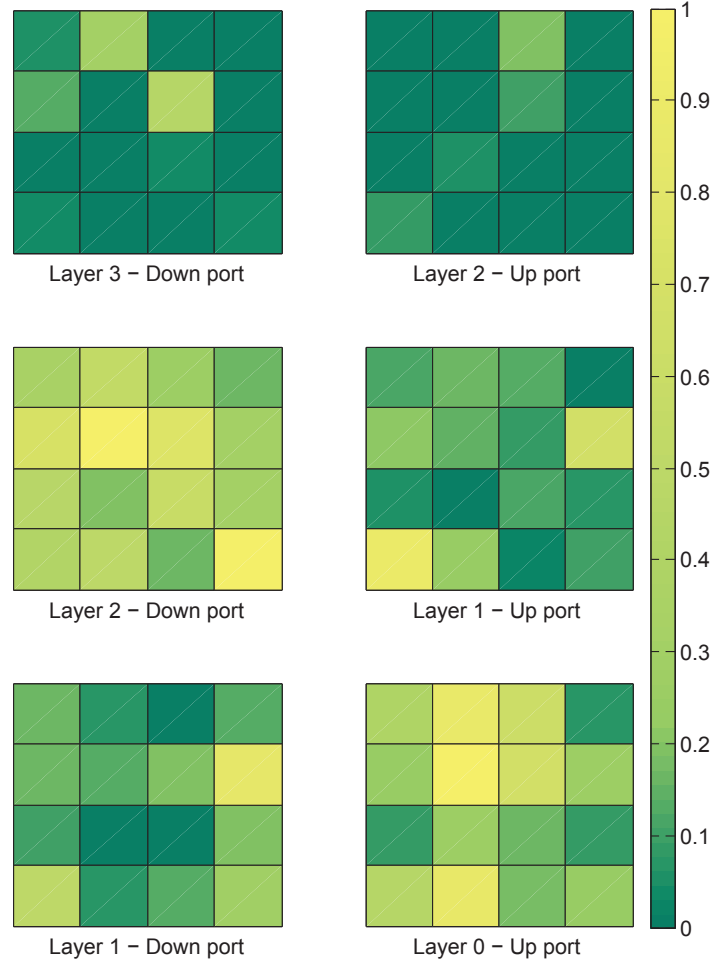


Figure 4.9: Fault density map

their design's sensitivity to TSV capacitive coupling faults under various TSV physical parameters and operating conditions.

## 4.5 Summary

In this chapter, a TSV-to-TSV capacitive coupling fault model was presented which can be easily deployed in system-level 3D-NoC simulators to detect the TSV-to-TSV

capacitive coupling fault at runtime as part of any dynamic fault injection process. The proposed model facilitates the exploration of resilient TSV-based 3D-NoCs and help researchers accurately evaluate their 3D designs when dealing with TSV capacitive coupling. The core of the fault model is implemented at the circuit-level to collect accurate timing violations for each of parasitic capacitive coupling cases, although the interface is implemented at the system-level. This model is useful for both application-specific and general purpose designs. For application specific designs, the fault model pinpoints the TSVs susceptible to failure. For general purpose applications, it can be applied to optimize the physical parameters of TSV to reduce the propagation delay of TSVs caused by TTCC.

## Part II

# TSV Coupling Mitigation

# Chapter 5

## Inductive TSV Coupling Mitigation

### 5.1 Introduction

As stated earlier, TSV coupling is one of the major issues in 3D-NoC designs because of increased parasitic signals as compared to 2D ICs, which may result in delay or even mutual coupling between adjacent TSVs [42, 62]. The term TSV coupling refers to capacitive and inductive couplings among neighbor TSVs, which the latter is more critical in higher frequency data transmissions [7].

In this chapter, the reliability of 3D-NoC against the inductive TSV-to-TSV coupling is investigated and a scalable inductive coupling aware coding is proposed. The proposed algorithm in this chapter is intended to support two major 3D device categories. The devised baseline algorithm [28] targets the first 3D device category which consists of designs with low TSV concentration (less than 100 TSVs) such as 3D NoC [111, 59, 60]. An enhanced scheme is proposed to support architectures with high



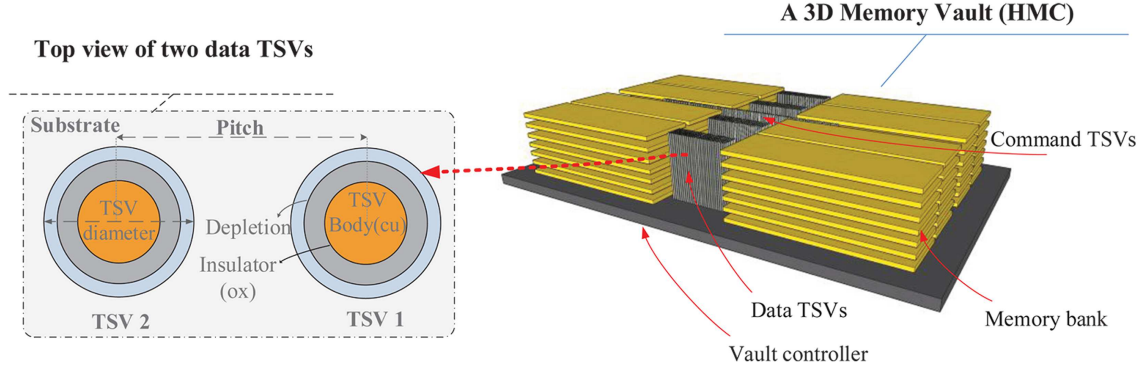


Figure 5.1: 3D-NoC vault, vertically interconnected by TSV bus in 3D integration technology

TSV concentration (around 500 TSVs) such as 3D DRAM memories (Hybrid Cube Memory (HMC) [45, 101, 73]), as shown in Figure 5.1. The experimental results show that the proposed coding algorithm yields significant improvements while its hardware-implemented encoder depicts tangible latency, power consumption, and area.

First, the reliability issue of inductive TSV-to-TSV coupling fault effect is analyzed within a 3D-NoC. Moreover, an analytical failure (data corruption) estimation of TSV links caused by inductive TSV coupling effect is presented. Subsequently, a method to minimize the effect of magnetic field caused by TSVs, including an analytical analysis to demonstrate the strength of proposed technique is devised. Also, a scalable coding approach with modest information redundancy overhead for implementing the proposed technique in large-scale 3D-NoCs is presented. Finally, the efficiency in mitigating TSV-to-TSV inductive coupling is demonstrated and the scalability and practicality of the proposed schemes is justified using concrete experiments and hardware implementation and synthesis.

## 5.2 Inductive TSV-to-TSV Coupling

The impact of TSVs on future 3D-ICs is still unknown [54]. However, chip warpage, TSV coupling, and thermal stress are known as main causes of TSV failure [105, 42].

The term TSV coupling refers to capacitive and inductive couplings among neighboring TSVs. Electric field results in capacitance coupling and magnetic field is a source of inductive coupling. Inductive coupling among neighboring TSVs is more critical in higher frequency data transmissions [110], and long TSVs which is considered in this chapter. Processors with higher operating frequency are emerging as the process technology is scaling down; an example is the IBM 5.2GHz multiprocessor [108].

### 5.2.1 Inductive Coupling Characteristics

To characterize the effect of inductive coupling, a 3x3 matrix of TSVs is modeled at the circuit-level with Synopsys HSPICE. The middle TSV is assumed to be the victim and the other 8 are the aggressors. In simulations, the top end of each TSV is connected to the output of an inverter, which drives the input of another inverter connected to the bottom of the TSV. The coupled TSV structure is modeled based on [51, 62] as a lumped RLC circuit. The circuit is composed of a series TSV resistance  $R_{TSV}$  and inductance  $L_{TSV}$ , parallel silicon substrate resistance  $R_{si}$  and capacitance  $C_{si}$ , and silicon dioxide capacitance  $C_{ox}$  around the TSV. These components form the relation between the signal TSVs. The circuit element values are extracted from analytical equations based on the dimensions of the structure, such as oxide thickness  $t_{ox}$ , silicon

substrate height  $h_{si}$ , TSV radius  $r_{TSV}$ , and TSV pitch  $P_{TSV}$  and by material properties like dielectric constant  $\epsilon$  and resistivity  $\rho$ . The equations are as follows:

$$R_{si} = \rho_{si} \frac{\cosh^{-1} \left[ \frac{P_{TSV}}{2r_{TSV}} \right]}{\pi h_{si}} \quad (5.1)$$

where

$$\rho_{si} = 0.0012T^2 - 0.0352T + 10 \quad (5.2)$$

and

$$C_{si} = \epsilon_{si} \frac{\pi h_{si}}{\cosh^{-1} \left[ \frac{P_{TSV}}{2r_{TSV}} \right]} \quad (5.3)$$

$$C_{ox} = \epsilon_{ox} \frac{2\pi h_{si}}{\ln \left[ \frac{r_{TSV} + t_{ox}}{r_{TSV}} \right]} \quad (5.4)$$

where

$$\epsilon_{ox} = 0.016T + 3.6 \quad (5.5)$$

TSV inductance [103] is also derived by partial self-inductance and mutual inductance. Partial self-inductance depends on the diameter and length of TSV and is expressed as:

$$L_{TSV_{self}} = \frac{\mu_0 l_{TSV}}{2\pi} \left[ \ln \left( \frac{2l_{TSV}}{r_{TSV}} \right) - \frac{3}{4} \right] \quad (5.6)$$

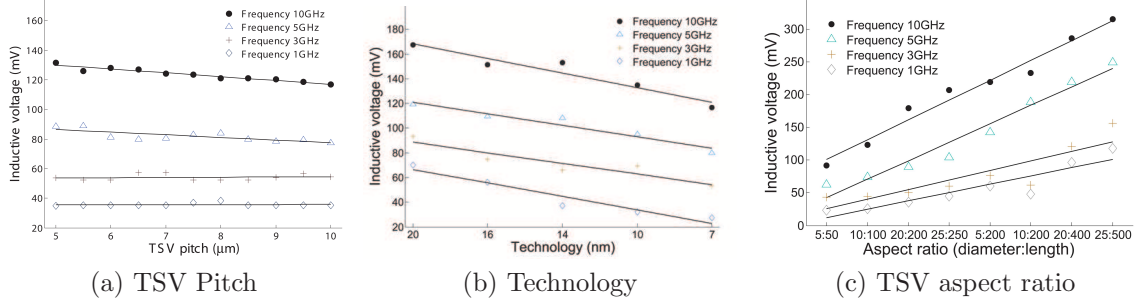


Figure 5.2: Inductive coupling SPICE simulation results

$$L_{TSV_{Mutual}} = \frac{\mu_0 l_{TSV}}{2\pi} \left[ \ln \left( \frac{l_{TSV}}{P_{TSV}} + \sqrt{1 + \left( \frac{l_{TSV}}{P_{TSV}} \right)^2} \right) - \sqrt{1 + \left( \frac{P_{TSV}}{l_{TSV}} \right)^2} + \frac{P_{TSV}}{l_{TSV}} \right] \quad (5.7)$$

where  $\mu_0$  is the permeability of free space given by  $4\pi \cdot 10^{-7}$ .

Predictive Technology Model (PTM) [88] FinFET transistor models are employed to implement inverters in this experiment. The worst-case induced voltage on the victim TSV is reported for different TSV pitches (Figure 5.2a), process technologies (Figure 5.2b), and TSV aspect ratio (Figure 5.2c) over different frequencies. The simulation parameter values are chosen according to ITRS [43] interconnect report, as shown in Table 5.1.

Based on Figure 5.2c, it is observed that as TSVs become longer (even with the same aspect ratio) the magnetic flux linking the two TSVs increases proportionally.

Table 5.1: Simulation configurations in Figure 5.2

<b>Figure</b>	<b>Technology</b>	<b>Length</b>	<b>Pitch</b>	<b>Diameter</b>
5.2a	14nm	100 $\mu m$	8 $\mu m$	4 $\mu m$
5.2b	20, 16, 14, 10, 7nm	100 $\mu m$	8 $\mu m$	4 $\mu m$
5.2c	14nm	10 – 500 $\mu m$	9 – 29 $\mu m$	5 – 25 $\mu m$

Therefore, as the length of TSVs grow, mutual coupling between aggressors and victim increases almost linearly and the coupled voltage rises proportionally.

Although the linkage flux between two TSVs is a strong function of the TSV length, its dependence on TSV-to-TSV pitch is weak. Changing the pitch between cylindrical TSVs affects mutual inductance in two ways. First, it changes the magnetic field created by the aggressor. Secondly, considering Faraday’s law, it alters the surface on which the magnetic field is integrated to calculate the linkage flux. As long as the proximity effect and other high order magnetic effects are trivial, current distribution in a TSV remains almost symmetrical regardless of the pitch size. Therefore, the magnetic field created by an aggressor does not vary by the pitch size, making the first effect to be negligible. Since the pitch between the TSVs is at least an order of magnitude smaller than their lengths, the second effect is small, but the linkage flux and consequently mutual coupling decreases slightly as pitch increases, shown in Figure 5.2a.

As shown in Figure 5.2b, induced voltage is a function of process technology. As processes advance, gate capacitance gets smaller and voltage rise/fall time becomes shorter. These two effects have opposite impacts on charging/discharging current of gate capacitance. The same current that charges (or discharges) the gate capacitance passes through TSV and causes inductive coupling to its neighboring TSVs. Thus, inductively coupled voltage varies for different technologies.

As technology advances and supply voltage shrinks, the coupled voltage becomes a greater portion of  $V_{dd}$ , resulting in higher probability of error. Among all the physical parameters, the length of the TSVs has the major impact on inductive coupling, specifically for global TSVs connecting more number of layers.

## 5.2.2 Problem Definition

An accurate analysis of coupling-induced failure requires a complex electromagnetic analysis of neighboring TSVs. Since such an analysis is outside the scope of this thesis, an approximate form of the problem is considered. Assuming the electromagnetic proximity effect and other high order effects can be neglected, the coupling-induced voltage  $\beta_{\text{tot}}$  is simply the sum of voltages induced on the victim TSV by the neighboring TSVs. Faraday's law implied as (Equation 5.8):

$$\beta_{\text{tot}} = \sum_{i=1}^N V_{\text{coupl},i} = \sum_{i=1}^N M_{v,i} \frac{dI_i}{dt} \quad (5.8)$$

where

- $N$  is the total number of aggressors.
- $V_{\text{coupl},i}$  is the voltage coupled on the victim by  $i^{\text{th}}$  aggressor, assuming all other aggressors have constant current.
- $M_{v,i}$  is the mutual inductance between  $i^{\text{th}}$  aggressor and victim TSVs.  $M_{v,i}$  is

calculated from Equation 5.9 [103].

$$M_{v,i} = \frac{\mu_0}{2\pi} \left[ l * \ln \left( \frac{l + \sqrt{d_i^2 + l^2}}{d_i} \right) + d_i - \sqrt{d_i^2 + l^2} \right] \quad (5.9)$$

where  $d_i$  is the distance of  $i^{th}$  aggressor from the victim TSV and  $l$  is the length of a TSV.

- $I_i$  is the current of  $i^{th}$  aggressor TSV.

The representation of  $\beta_{tot}$  can be further simplified as follows. Assume that the inductive coupling voltage caused by a single horizontal or vertical neighboring TSV is  $\beta_{tot}$ , then  $\beta_{tot}$  is equal to  $\alpha \times \beta$ , where the parameter  $\alpha$  depends on the current flow direction and arrangement of active neighboring TSVs.

Each victim TSV has four neighbors in horizontal or vertical directions. Figure 2.5, (shown earlier in Chapter 2), depicts a top view of different geometrical possibilities of neighbor configurations. Only 4 neighbors are considered to simplify the analysis. The parameter  $\alpha$  equals the algebraic sum of current values in neighbors of the victim TSV. With only 4 neighbors,  $\alpha$  assumes a values in  $\{-4, \dots, 4\}$ . Clearly, the severity of inductive coupling is higher for larger absolute values of  $\alpha$  and the goal of this chapter is to reduce the current configurations that lead to high values of  $|\alpha|$ . In other words, the higher the sum of neighboring currents, the higher is the inductively coupled voltage, and higher vulnerability to error.

With the cross section view of TSVs, the current of horizontal and vertical neighbors of a victim TSV are examined to measure the severity of inductive TSV-to-TSV coupling. The effect of diagonal neighboring TSVs, which causes less mutual coupling

effect than adjacent TSVs, is not considered in this work.

### 5.2.3 Coding Scheme

The main contribution of this chapter is to propose a coding scheme to mitigate inductive coupling occurrences by adjusting the sequence of data flits<sup>1</sup>. The suggested coding technique replaces larger  $\alpha$  values by smaller ones. While the baseline algorithm is intended to show the mitigation gain obtained by using the proposed method, a variation of the baseline algorithm called “enhanced algorithm” introduces scalability into the baseline approach.

One possible approach to data manipulation is to perform inversion on a properly-selected set of input bit streams. For this method, decoding of the received signal requires knowledge of the location of the bits that have been inverted, inflicting serious overhead. One workaround for this overhead is to perform inversion on TSV data rows rather than individual TSV bits. Clearly, this reduction in overhead comes at the cost of inferior gain.

The outline of baseline algorithm is explained as a two-phase algorithm:

1. In the first phase, each cell decides whether or not to submit inversion requests to its vertical neighbors (above and below) with the goal of decreasing the sum of its neighboring currents. Submitting the requests to only two neighbors, rather than four neighbors, is chosen for the sake of simplicity of the design. These requests are stored for the future processing.

---

<sup>1</sup>flit stands for “flow control unit”



2. Once all requests have been submitted, cells process their received requests and decide whether or not to accept inversion. Finally, based on these individual decisions, each row decides whether or not to grant inversion.

### 5.2.4 System Model

Assume that the number of bits to be transmitted over TSVs is denoted by  $L_D$  at each time slot. The encoded data block is sent over a matrix of TSVs with  $N_R$  rows and  $N_C$  columns, with  $N_R$  and  $N_C$  satisfying  $N_R \times N_C = L_D$ .

With this convention, the original data to be transmitted at time slot  $t$  is represented by  $D_t$  matrix. Similarly, the encoded data that has already been sent at time  $t - 1$  is represented by  $\hat{D}_{t-1}$ .

The current flow direction of each TSV is specified by the modified data already sent over the TSV, namely  $\hat{d}_{t-1}$  ( $\hat{d}_{t-1}$  is a cell of  $\hat{D}_{t-1}$  matrix). Similarly,  $\hat{d}_t$  represents the encoded data bit to be transmitted, while  $d_t$  means the original bit to be transmitted at time slot  $t$ . With this convention and the proposed inversion mechanism,  $\hat{d}_t$  will be either  $d_t$  or  $\bar{d}_t$ . A simple analysis of the circuitry connected to a TSV reveals that:

1. If the  $\hat{d}_{t-1} = 0$  and  $\hat{d}_t = 0$ , the TSV current will be 0 ( $\odot$ ).
2. If the  $\hat{d}_{t-1} = 0$  and  $\hat{d}_t = 1$ , the TSV current will be 1 ( $\odot$ ).
3. If the  $\hat{d}_{t-1} = 1$  and  $\hat{d}_t = 0$ , the TSV current will be -1 ( $\otimes$ ).
4. If the  $\hat{d}_{t-1} = 1$  and  $\hat{d}_t = 1$ , the TSV current will be 0 ( $\odot$ ).

with  $\odot$ ,  $\circ$ , and  $\otimes$  representing current values of 1, 0, and -1, respectively. Consequently, the direction of current can be calculated, if  $\hat{d}_{t-1}$  and  $\hat{d}_t$  are known. From the preceding discussion, it is easy to see that the  $N_R \times N_C$  matrix  $C$  representing the current of TSVs is equal to:

$$C(D1, D2) = D2 - D1 \quad (5.10)$$

The key parameter in baseline algorithm is the sum of neighboring TSV current. Therefore, it is helpful to define an  $N_R \times N_C$  matrix  $P$ , where the  $(i, j)^{th}$  element of  $P$  ( $P_{ij}$ ) is equal to algebraic sum of neighboring TSV currents. From this definition, the elements of  $P$  can take any values in the set  $\{-4, \dots, 4\}$ .

## 5.3 Baseline Coding Algorithm

In the proposed coding, each cell (corresponding to each TSV) sends/receives an inversion request to the cell above or below itself, based on its neighbor condition. These neighbor cells then decide, based on the received requests, whether or not to honor the requests. Before delving into the details, the effect of bit inversion on the TSV current should be examined.

### 5.3.1 Effect of bit inversion on TSV current

The direction of current passing through a TSV is specified by the previous data bit which is already sent over the TSV and the data bit to be sent over the TSV, as

discussed in Section 2.5.1. In the coding algorithm,  $\hat{d}_t = \bar{d}_t$ , if the inversion decision is taken, otherwise  $\hat{d}_t = d_t$ . The change of current is summarized as follows:

1. In case of no inversion,  $(\hat{d}_{t-1}, \hat{d}_t = d_t) = (0, 0)$  results in  $\circ$ . If an inversion is performed, i.e.  $(\hat{d}_{t-1}, \hat{d}_t = \bar{d}_t) = (0, 1)$ , the current will be  $\odot$ .
2. In case of no inversion,  $(\hat{d}_{t-1}, \hat{d}_t = d_t) = (0, 1)$  results in  $\odot$ . If an inversion is performed, i.e.  $(\hat{d}_{t-1}, \hat{d}_t = \bar{d}_t) = (0, 0)$ , the current will be  $\circ$ .
3. In case of no inversion,  $(\hat{d}_{t-1}, \hat{d}_t = d_t) = (1, 0)$  results in  $\otimes$ . If an inversion is performed, i.e.  $(\hat{d}_{t-1}, \hat{d}_t = \bar{d}_t) = (1, 1)$ , the current will be  $\circ$ .
4. In case of no inversion,  $(\hat{d}_{t-1}, \hat{d}_t = d_t) = (1, 1)$  results in  $\circ$ . If an inversion is performed, i.e.  $(\hat{d}_{t-1}, \hat{d}_t = \bar{d}_t) = (1, 0)$ , the current will be  $\otimes$ .

It is concluded that  $\circ$  can be changed to both  $\otimes$  and  $\odot$  (If  $d_t=1$  or  $0$  respectively) by the inversion of  $d_t$ , while  $\odot$  and  $\otimes$  are only changed to  $\circ$ .

### 5.3.2 Reducing the sum of neighbor currents by inversion

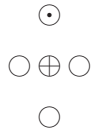
The proposed coding consists of two phases as follows:

#### Submitting inversion requests

As mentioned previously, the goal of each cell is to see how it can reduce the sum of its neighbors' current by inverting the data on neighbor TSVs above and below itself. Table 5.2 lists all possible forms of requests that can be submitted by a victim TSV

$\oplus$  to its neighbors to reduce  $|\alpha|$ . The classification is based on various scenarios that happen for the neighbors above and below of a victim TSV. The proposed actions in Table 5.2 are based on two factors; first, the upward current flow ( $\odot$ ) conversion to downward one ( $\otimes$ ) is not possible. Second, the current change, achieved by inversion of corresponding neighbor, should be adjusted in such a way that the magnitude of the sum of neighbor currents decreases. It is easy to verify that the proposed actions in the table follow these guidelines.

In some of the configurations of Table 5.2, requests are sent to only one of the vertical neighbors, while in others, the requests are sent to both neighbors. The former is identified by the word 'only' in the third column of Table 5.2. The following example illustrates the necessity of sending a single request to only one of the neighbors. Consider the following TSV current configuration:



If the neighbor above changes as  $\odot \rightarrow \bigcirc$ , the sum of currents will be 0 which is the ideal situation. In a similar way, the same result holds if only the neighbor below changes as  $\bigcirc \rightarrow \otimes$ .

It is desirable to have a simple decision rule rather than a lookup table in order to figure out when to submit an inversion request. The entire set of proposed requests in Table 5.2 is summarized in a very simple form.

1. For a cell  $(i, j)$  with  $P[i][j] > 0$ , if the data of target neighbor is 1 and current of that neighbor is not -1 ( $\otimes$ ), send an inversion request to that neighbor.

Table 5.2: Reducing the sum of neighbor currents by inverting vertical neighbors

$ \alpha $	Typical patterns	Inversion request to vertical neighbors
1	$\begin{array}{c} \circ \\ \odot \oplus \circ \\ \circ \end{array}$	only one $\circ \rightarrow \otimes$
	$\begin{array}{c} \odot \\ \circ \oplus \circ \\ \circ \end{array}$	only one of these: $\circ \rightarrow \otimes$ or $\odot \rightarrow \circ$
	$\begin{array}{c} \otimes \\ \odot \oplus \odot \\ \circ \end{array}$	$\circ \rightarrow \otimes$
	$\begin{array}{c} \odot \\ \otimes \oplus \odot \\ \circ \end{array}$	only one of these: $\circ \rightarrow \otimes$ or $\odot \rightarrow \circ$
	$\begin{array}{c} \odot \\ \odot \oplus \circ \\ \otimes \end{array}$	$\odot \rightarrow \circ$
	$\begin{array}{c} \odot \\ \otimes \oplus \circ \\ \odot \end{array}$	only one $\odot \rightarrow \circ$
2	$\begin{array}{c} \odot \\ \circ \oplus \circ \\ \odot \end{array}$	$\odot \rightarrow \circ$
	$\begin{array}{c} \odot \\ \odot \oplus \circ \\ \circ \end{array}$	$\odot \rightarrow \circ$ and $\circ \rightarrow \otimes$
	$\begin{array}{c} \circ \\ \odot \oplus \odot \\ \circ \end{array}$	$\circ \rightarrow \otimes$
	$\begin{array}{c} \odot \\ \odot \oplus \otimes \\ \odot \end{array}$	$\odot \rightarrow \circ$
	$\begin{array}{c} \odot \\ \odot \oplus \odot \\ \otimes \end{array}$	$\odot \rightarrow \circ$
3	$\begin{array}{c} \odot \\ \odot \oplus \circ \\ \odot \end{array}$	$\odot \rightarrow \circ$
	$\begin{array}{c} \odot \\ \odot \oplus \odot \\ \circ \end{array}$	$\odot \rightarrow \circ$ and $\circ \rightarrow \otimes$
4	$\begin{array}{c} \odot \\ \odot \oplus \odot \\ \odot \end{array}$	$\odot \rightarrow \circ$

- For a cell  $(i, j)$  with  $P[i][j] < 0$ , if the data of target neighbor is 0 and current of that neighbor is not 1 ( $\odot$ ), send an inversion request to that neighbor.

The result of such decision is stored in two  $N_R \times N_C$  matrices, called “Request From Above” ( $RFA$ ) and “Request From Below” ( $RFB$ ). The elements of these matrices are either 0 or 1. If  $RFA[i][j]$  is 1, it means that the cell at  $(i, j)$  has received an inversion request from the cell above itself.  $RFB$  is defined similarly.  $RFA$  and  $RFB$  are initialized to 0 before any operation. Also, note that the first (last) row does not receive any requests from above (below).

### Processing inversion requests

Once all inversion requests are submitted and cells with mutually exclusive requests are marked, the inversion decision is made. One possible technique for cell  $(i, j)$  is to grant such an inversion request only if  $RFA[i][j] = 1$  and  $RFB[i][j] = 1$ . Since the top row and bottom row do not have any neighbors above and below them respectively, the first row of  $RFA$  and the last row of  $RFB$  are set to 1 in order to avoid decision conflicts with the proposed approach.

Once  $RFA$  and  $RFB$  are finalized, they are combined as an intention matrix  $I^{\text{mat}} = \text{AND}(RFA, RFB)$ . If  $I^{\text{mat}}[i][j] = 1$ , the data bit of cell  $(i, j)$  is marked for inversion. Since the final inversion is row-based rather than cell based, an intention vector  $I^{\text{vec}}$  (of size  $N_R \times 1$ ) is constructed from  $I^{\text{mat}}$ .

If the number of 1’s are greater than the number of 0’s, row  $i$  is selected to be inverted, and  $I^{\text{vec}}[i]$  is set to 1.  $I^{\text{vec}}$  is initialized to zero.

---

**Algorithm 5.1.** Summary of baseline algorithm

---

- 1: Take  $\hat{D}_t$  and  $D_t$  as inputs
  - 2: Construct  $C$  by  $C = D_t - \hat{D}_{t-1}$
  - 3: Construct matrix  $P$  by setting its  $(i, j)^{th}$  element  $P[i][j]$  equal to the algebraic sum of the currents of neighbors of TSV  $(i, j)$
  - 4: Initialize  $RFA$  and  $RFB$  to 0. Then set the first row of  $RFA$  and the last row of  $RFB$  to 1
  - 5: **for** all  $i$  and  $j$ ,  $i \neq N_R$ , set  $RFB[i][j] = 1$  if the OR of following conditions are true **do**
  - 6:      $P[i+1][j] < 0$  and  $D_t[i][j] == 0$
  - 7:      $P[i+1][j] > 0$  and  $D_t[i][j] == 1$
  - 8: **end for**
  - 9: **for** all  $i$  and  $j$ ,  $i \neq 1$ , set  $RFA[i][j] = 1$  if at least one of the following conditions holds **do**
  - 10:      $P[i-1][j] < 0$  and  $D_t[i][j] == 0$
  - 11:      $P[i-1][j] > 0$  and  $D_t[i][j] == 1$
  - 12: **end for**
  - 13:  $I^{mat} = \text{AND}(RFA, RFB)$
  - 14:  $I^{vec}[i] = \sum_{j=1}^{N_C} I^{mat}[i][j]$  for all  $i \in \{1, \dots, N_R\}$ .
  - 15:  $I^{vec}[i] = \mathbf{1}(I^{vec}[i] \geq N_C/2)$ , where the identity function  $\mathbf{1}$  returns 1 when its argument is true, and returns 0 otherwise.
  - 16:  $\hat{D}_t[i][1\dots N_C] = \text{XOR}(I^{vec}[i], D_t[i][1\dots N_C])$  for all  $i$ 's
- 

### 5.3.3 An Example of Baseline Algorithm

Suppose that the previously transmitted data and the unmodified current data are:

$$\hat{D}_{t-1} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, D_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

If  $D_t$  is transmitted, the current matrix and the corresponding  $P$  matrix will be:

$$C = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \odot & \otimes & \otimes & \otimes \\ \circ & \circ & \odot & \odot \\ \circ & \circ & \circ & \circ \\ \otimes & \odot & \odot & \circ \end{bmatrix}$$

$$P = \begin{bmatrix} -1 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ -1 & 1 & 2 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

In this case, the number of cells with  $P = 0$  through  $P = 4$  is 6, 9, 1, 0, and 0, respectively. Now, suppose that the baseline algorithm is applied to  $(D_t, \hat{D}_{t-1})$ .  $RFA$  and  $RFB$  will hold the following values:

$$RFB = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, RFA = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$



Then, the  $I^{\text{mat}}$  and the corresponding  $I^{\text{vec}}$  are:

$$I^{\text{mat}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, I^{\text{vec}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Which results in the inversion of last row of  $D_t$ :

$$\hat{D}_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

With this inversion, the new current and  $P$  matrix will be:

$$C = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, P = \begin{bmatrix} -1 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Note that after inversion, the count of cells with  $P = 0$  through  $P = 4$  is 11, 5, 0, 0, and 0, respectively: While the number of cells with  $P = 0$  has grown from 6 to 11, the number of cells with  $P = 1$  and  $P = 2$  has dropped from 9 to 5 and from 1 to 0.

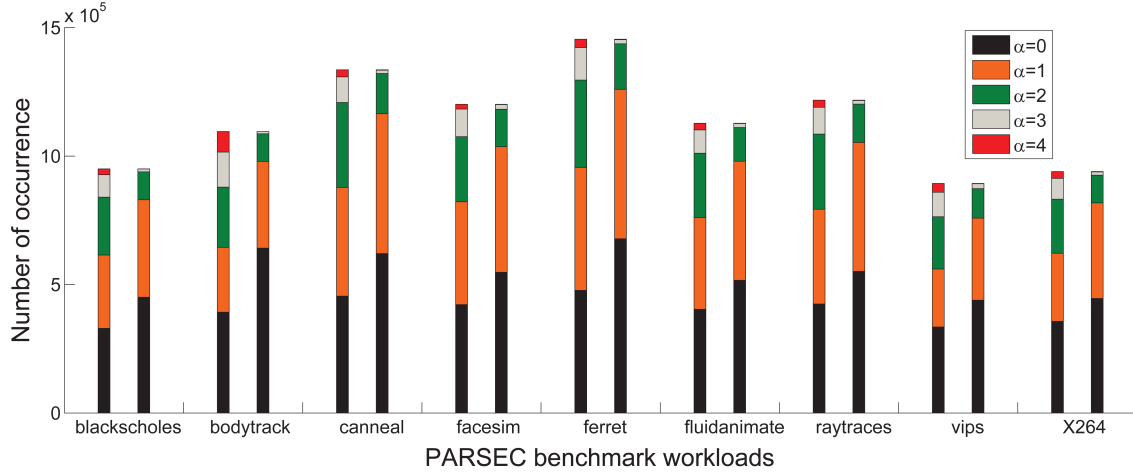


Figure 5.3: Evaluating the efficiency of baseline algorithm for an  $8 \times 8$  TSV bus with PARSEC data traffic. For each workload the left bar represents the uncoded and the right bar shows the coded approach results.

## 5.4 Baseline Algorithm Evaluation

In order to evaluate the efficiency of the proposed code, a long random sequence of bits is input into two systems: one with encoder, and the other without encoder. Also, in order to have a comprehensive evaluation of the baseline algorithm, PARSEC [11] benchmark memory traces captured using PIN instrumentation tool [39] are utilized as a real-world data traffic. Denote the class of cells with  $|P_{ij}| = 0, \dots, 4$  by  $\psi_k$  for  $k = 0, \dots, 4$ . Then, the relative occurrence frequency (denoting by  $F(i)$ ) of all  $\psi_i$ 's are counted for different  $i$ 's. Comparing the occurrence frequency diagrams of the two systems shows whether the proposed coding lowers the occurrence of  $\psi_i$ 's with larger  $i$ . If such a decline is observed, the result is a decrease in inductive coupling, as discussed earlier.

Figure 5.4a and Figure 5.3 show the relative occurrence of different  $\psi_i$ s for an

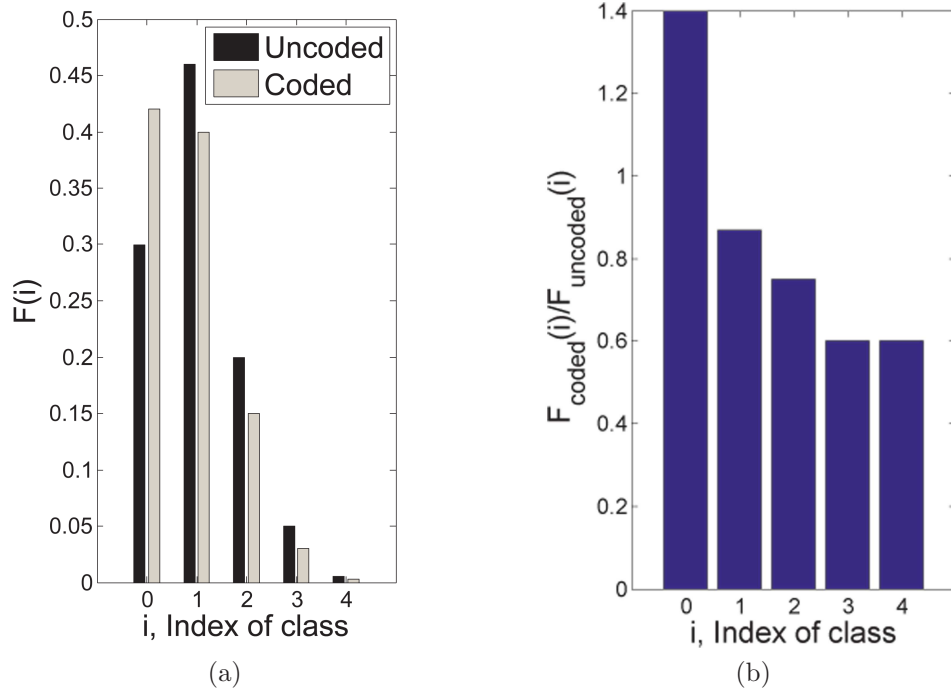


Figure 5.4: Evaluating the efficiency of baseline algorithm for an  $8 \times 8$  TSV bus with random data traffic.

$8 \times 8$  TSV bus in both uncoded (left bar) and coded (right bar) system with random and PARSEC benchmark data traffic, respectively. In Figure 5.4a, there are 5 pairs of columns, where the left column of each column pair belongs to the uncoded system and the right column belongs to the coded system.

As it can be observed in Figure 5.4, the relative frequency of occurrence of  $\psi_i$ 's with large  $i \geq 2$  has decreased. Furthermore, the relative frequency of occurrence of  $\psi_i$  remains almost the same. Importantly, the relative frequency of occurrence of  $\psi_0$  has increased. Figure 5.4b depicts the ratio of right columns to the left column of column pairs of Figure 5.4a, to emphasize the change of relative frequency of occurrence of different  $\psi_i$ 's.

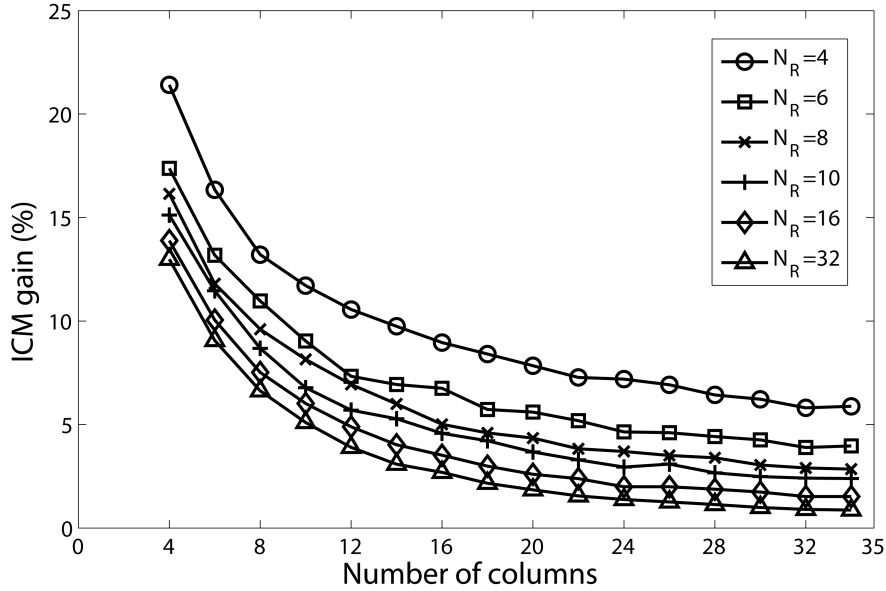


Figure 5.5: ICM gain versus number of columns

### 5.4.1 Evaluation Metrics

Apart from the visual conclusion, it is possible to define a scalar measure to quantify the effectiveness of the proposed coding. One possible solution is to form a weighted sum of relative frequencies, where the weight of the relative frequency of  $\psi_i$  is taken to be  $i$ , to impose a penalty on high values of  $i$ . Consequently, the lower the value of this measure, the better is the efficiency. Inductive Coupling Mitigation (ICM) metric is defined, as indicated by  $\mu$  in Equation 5.11, to evaluate the efficiency of the proposed algorithm.

$$\mu = \sum_{i=0}^4 i.F(i) \tag{5.11}$$

The “ICM gain” of the coding algorithm is also quantified as the ratio of “change

in  $\mu$  due to coding” to the “ $\mu$  of uncoded system”:

$$\nu = \frac{|\mu_{\text{uncoded}} - \mu_{\text{coded}}|}{\mu_{\text{uncoded}}} \quad (5.12)$$

Applying this measure to the result of Figure 5.4 shows that the ICM measure changes from 0.99 to 0.78, which results in a 21% ICM gain.

Finally, the proposed coding imposes an overhead for data transmission. In order to transmit  $L_D$  bits over an  $N_R \times N_C$  TSV matrix ( $L_D = N_R \times N_C$ ), an additional  $N_R$  bits are required to transmit  $I^{\text{vec}}$  alongside the modified data. Thus, the overhead is written as:

$$\eta = \frac{N_R}{N = N_R N_C} = \frac{1}{N_C} \quad (5.13)$$

## 5.4.2 Scalability of Baseline Algorithm

It is of interest to analyze the variation of ICM gain  $\nu$  with TSV bus dimensions  $N_R$  and  $N_C$ . This analysis provides an efficient physical distribution and placement of TSVs within a vault. For a given bus size  $N_R \times N_C$  and under certain constraints on  $\nu$  and  $\eta$ , this information is used to decide on the values of  $N_R$  and  $N_C$ .

Figure 5.5 and Figure 5.6 show the gain improvement for the fixed number of rows (columns) as the number of columns (rows) grows. It is observed that increasing both  $N_R$  and  $N_C$  decreases  $\nu$ ; however, the negative effect of increasing  $N_C$  is more severe than the effect of increasing  $N_R$ . Figure 5.7 illustrates an instance of this fact by comparing  $\nu$  for the two cases of  $N_R = 4$  and  $N_C = 4$ . It is observed that at each

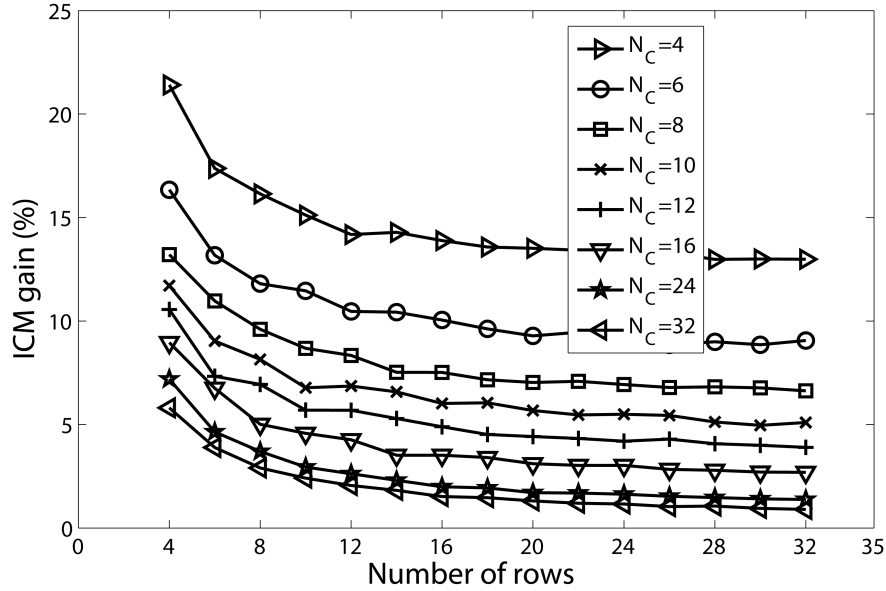


Figure 5.6: ICM gain versus number of rows

fixed bus size, the ICM gain is better when the number of columns has a fixed value.

In the following, the observed variation of  $\nu$  with  $N_C$  and  $N_R$  are justified. A rigorous justification of the variation of  $\nu$  with  $N_C$  and  $N_R$  requires calculating the values of  $F(i)$  [for  $i = 0, \dots, 4$ ] for the coded system. However, this calculation requires analyzing a very large scale Markov chain which is outside the scope of this thesis. Consequently, the indirect methods are used to justify the observed behavior.

### Justification of $N_C$ effect on ICM gain

An indirect approach for examining the impact of  $N_C$  on  $\nu$  is to calculate the probability of row inversion. Given the algorithm, it is reasonable to assume that adjacent

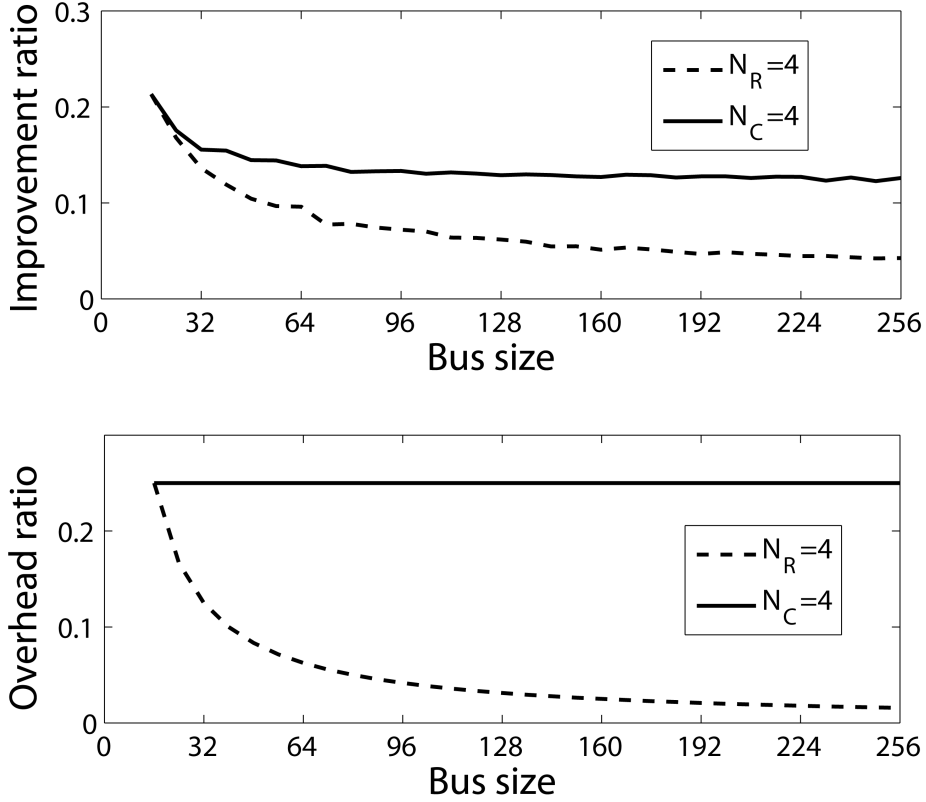


Figure 5.7: ICM gain and corresponding information redundancy for the same number of bits in a bus with column or row growth

cells in a row are marked independently for inversion, i.e.:

$$\mathcal{P}(I_{i,j}^{\text{mat}} = 1 \mid \{I_{i,n}^{\text{mat}}, n \in \{1, \dots, N_C\} \setminus \{j\}\}) \approx \mathcal{P}(I_{i,j}^{\text{mat}} = 1)$$

where  $\mathcal{P}$  represents probability of its argument, “ $\setminus$ ” stands for set complement operator, and  $I_{i,j}^{\text{mat}}$  denotes the element of  $I^{\text{mat}}$  at row  $i$  and column  $j$ . With the assumption of same inversion probability for all cells (data bits)  $\mathcal{P}_{\text{inv,cell}}$ , and given the fact that a row is inverted only when more than half of its cells are marked for inversion, the

probability of row inversion is represented by Equation 5.14.

$$\mathcal{P}_{\text{inv,row}} = \sum_{i=\lceil N_c/2 \rceil}^{N_c} \binom{N_c}{i} (\mathcal{P}_{\text{inv,cell}})^i (1 - \mathcal{P}_{\text{inv,cell}})^{N_c-i} \quad (5.14)$$

Figure 5.8 shows  $\mathcal{P}_{\text{inv,row}}$  as a function of  $N_C$  for different values of  $\mathcal{P}_{\text{inv,cell}}$ . It is observed that when  $\mathcal{P}_{\text{inv,cell}} < 0.5$ ,  $\mathcal{P}_{\text{inv,row}}$  decreases with  $N_C$ . Combining this observation with the fact that  $\mathcal{P}_{\text{inv,cell}} < 0.5$  for baseline algorithm, it is concluded that increasing  $N_C$  decreases  $\mathcal{P}_{\text{inv,row}}$  within the proposed algorithm. As  $\mathcal{P}_{\text{inv,row}}$  decreases, the code is less frequently employed, and this reluctance for engaging the inversion mechanism deprives the system of the ICM gain promised by coding. Consequently, the observed descending trend of ICM gain with the increase of  $N_C$  is justified.

In practice, those cells that are closer to the border and corners have higher probability of inversion. To elaborate,  $\mathcal{P}_{\text{inv,cell}}$  is a function of probability of 0's and 1's in the input bit stream and the location of the cell. Consequently, the calculation of row inversion probability in Equation 5.14 is not accurate enough. However, as long as the maximum cell inversion probability of the cells in a row is less than 0.5, the same descending trend of row inversion probability with  $N_C$  is observed (see Figure 5.8).

### Justification of $N_R$ effect on ICM gain

The effect of  $N_R$  on  $\nu$  is examined in a similar fashion. The highest probability of row inversion belongs to rows 1 and  $N_R$ , while rows 2 and  $N_R - 1$  have lower probability of inversion, and the lowest probability belongs to rows 3, 4,  $\dots$ ,  $(N_R - 2)$ . Denote these probabilities by  $\mathcal{P}_{\text{inv,row}}^{(1,N_R)}$ ,  $\mathcal{P}_{\text{inv,row}}^{(2,N_R-1)}$ , and  $\mathcal{P}_{\text{inv,row}}^{(3 \dots N_R-2)}$ . Assuming that rows are inverted



independently, the average of the ratio of inverted rows (for  $N_R \geq 4$ ) to  $N_R$  is shown in Equation 5.15.

$$\overline{N_{\text{inv}}} = \frac{2}{N_R} \mathcal{P}_{\text{inv,row}}^{(1,N_R)} + \frac{2}{N_R} \mathcal{P}_{\text{inv,row}}^{(2,N_R-1)} + \left(1 - \frac{4}{N_R}\right) \mathcal{P}_{\text{inv,row}}^{(3 \dots N_R-2)} \quad (5.15)$$

By calculating the sensitivity of  $\overline{N_{\text{inv}}}$  to  $N_R$ , i.e.  $(d\overline{N_{\text{inv}}}/dN_R)/(\overline{N_{\text{inv}}})$ , it is possible to indirectly justify the effect of  $N_R$  on  $\nu$ :

$$\frac{d\overline{N_{\text{inv}}}/dN_R}{\overline{N_{\text{inv}}}} = \frac{\frac{-2}{N_R^2} \mathcal{P}_{\text{inv,row}}^{(1,N_R)} + \frac{-2}{N_R^2} \mathcal{P}_{\text{inv,row}}^{(2,N_R-1)} + \frac{4}{N_R^2} \mathcal{P}_{\text{inv,row}}^{(3 \dots N_R-2)}}{\frac{2}{N_R} \mathcal{P}_{\text{inv,row}}^{(1,N_R)} + \frac{2}{N_R} \mathcal{P}_{\text{inv,row}}^{(2,N_R-1)} + \left(1 - \frac{4}{N_R}\right) \mathcal{P}_{\text{inv,row}}^{(3 \dots N_R-2)}} = \frac{1}{N_R} \mathcal{O}(1) \quad (5.16)$$

where  $\mathcal{O}(1)$  represents a scalar of order 1. For  $N_R > 10$ , this ratio is very small and keeps getting smaller for large  $N_R$ . This reduction in sensitivity means that the average fraction of inverted rows remains almost constant and the ICM gain does not decrease significantly. This justifies why increasing the number of rows does not have a noticeable impact on the ICM gain.

While the discussion so far suggests that a minimal value of  $N_C$  is beneficial as long as the only parameter of interest is the ICM gain  $\nu$ , low values of  $N_C$  lead to a higher value of overhead, since the overhead is given by  $1/N_C$ .

## 5.5 Enhanced Coding Algorithm

In some 3D-IC devices, a large bus size and high ICM gain are required simultaneously. 3D stacked-DRAM is an example, in which the size of TSV data bus typically

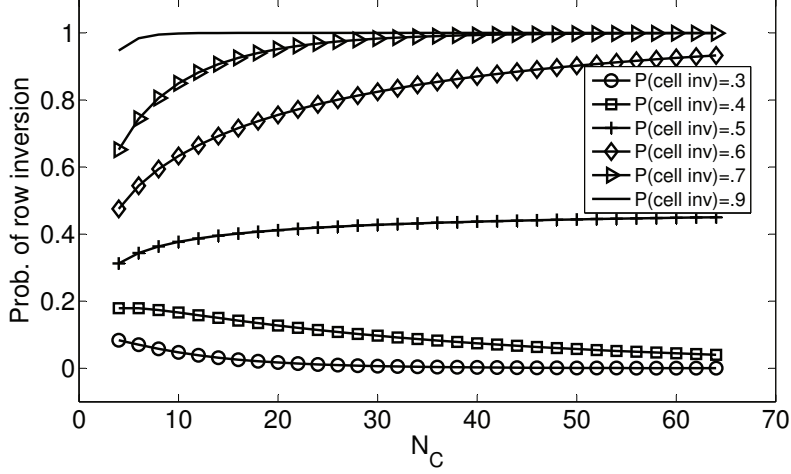


Figure 5.8: Probability of row inversion versus number of cells in a row

exceeds hundred bits. As discussed before, the ICM gain drops as  $N_R$  or  $N_C$  increases. This means that the proposed coding is not scalable and may not be able to provide sufficient ICM gain for applications such as 3D stacked-DRAM with large bus sizes. A partitioning scheme is proposed in the following section to make the ICM gain of the proposed coding scalable for larger bus sizes.

### 5.5.1 Partitioning Approach

Given that the ICM gain of baseline algorithm is significant for a small bus size, in this chapter an approach is proposed to make a large bus size gain-scalable by partitioning the large  $N_R \times N_C$  matrix of TSVs into  $q \times p$  sub matrices of size  $N_R/q$  and  $N_C/p$  and apply the coding on all sub matrices independently, resulting in an enhanced algorithm. With this method, the ICM gain of  $N_R \times N_C$  network is equal to the ICM gain of an  $(N_R/q) \times (N_C/p)$  network at the cost of increased information overhead. Denoting the ICM gain and overhead of an  $m \times n$  with a  $q \times p$  partitioning, the new

$\nu(m, n, q, p)$  and  $\eta(m, n, q, p)$  are presented by Equation 5.17:

$$\begin{aligned}\nu(m, n, q, p) &= \nu\left(\frac{m}{q}, \frac{n}{p}, 1, 1\right) \\ \eta(m, n, p, q) &= \frac{p \cdot q \cdot \frac{N_R}{q}}{N_R \cdot N_C} = \frac{p}{N_C} = p \cdot \eta(m, n, 1, 1)\end{aligned}\tag{5.17}$$

A column partitioning is considered here for the following reasons, i.e  $q = 1$ , instead of row or row-column partitioning. First, the impact of  $N_C$  on  $\nu$  is more severe than the effect of  $N_R$  on  $\nu$ , as discussed in Section 5.4.2. Therefore, performing partitioning on columns is preferable. Second, row partitioning does not offer substantial gain improvement because of the insensitivity of  $\nu$  to  $N_R$ . In column partitioning approach, the same ICM is obtained as reported for a small matrix while the overhead is the same as that of increasing rows.

A general benefit of partitioning is to make parallel implementation possible, resulting in a faster encoder architecture. The factor of parallelism is equal to the number of partitions.

### 5.5.2 Enhanced Algorithm Evaluation

The ICM gain and overhead of enhanced and baseline algorithms are compared in Figure 5.9 and Figure 5.10. The row scaling approach (constant  $N_C$ ) offers a consistent ICM gain, but suffers from a constant high amount of overhead regardless of the bus size. On the other hand, the column scaling approach (constant  $N_R$ ) improves overhead as bus size increases, but causes a dramatic decline in the ICM gain. The enhanced

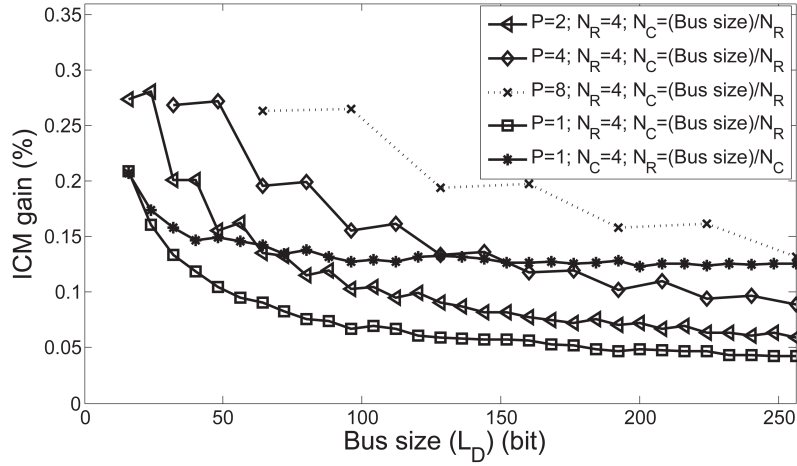


Figure 5.9: ICM gain for the same bus size with different partitions ( $P$ ) vs row ( $N_R$ ) growth

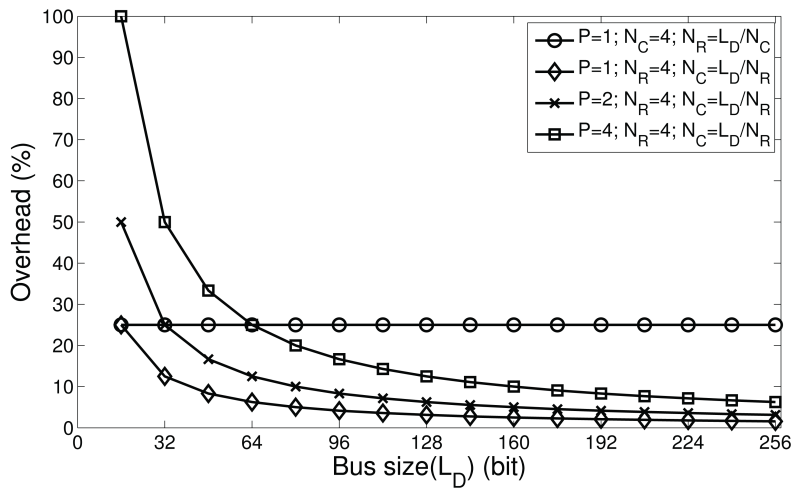


Figure 5.10: Information redundancy overhead rate for the same bus size with different partitions ( $P$ ) vs row ( $N_R$ ) growth

algorithm combines the high ICM gain of row-scaling approach with the low overhead of column scaling approach.

### 5.5.3 Hardware Synthesis Results

The encoder of both proposed coding algorithms is synthesized by Synopsys Design Compiler using 28nm TSMC library (1.05V, 25 °C) to report latency, power consumption, and area. Decoder component is the same for both of the proposed baseline and enhanced algorithms. The overhead is also negligible as compared to the encoder unit in terms of latency, power consumption, and area, since it is composed of a simple comparator and a mix of inverter gates.

A TSV data bus with variable size is also modeled in HSPICE. The TSV model in [62] is used to capture the inductive TSV-to-TSV coupling effect.

At each step of hardware implementation, inversion requests of a single TSV row are processed. Therefore, the latency of the encoder is constrained by TSV row size. The latency of row scaling is observed to be higher than the latency of column scaling, as illustrated in Table 5.3. In partitioning method, the number of columns in a partition decreases as the number of partitions increases, reducing the overall processing latency. This is because all the smaller partitions are evaluated in parallel. For example, for a 512 TSV bundle, the latency of encoder with 8 partitions is almost 50% of the one with one partition.

A larger bus size needs more complex combinational logic and memory units, resulting in higher power consumption and area footprint, as depicted in Figure 5.11 and Figure 5.12. However, partitioning the columns of TSVs, the hardware complexity declines as fewer net wirings and logic components are needed. Figure 5.11 demonstrates the decreasing trend in area for the same bus size, as the partitioning factor grows.

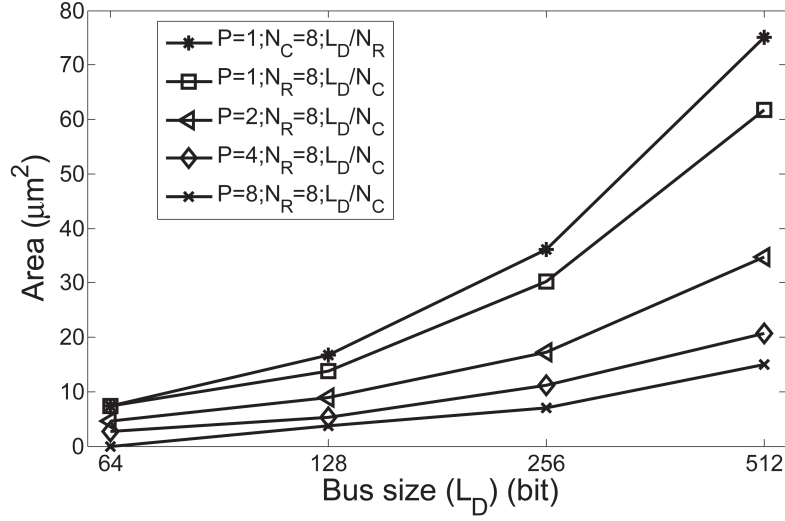


Figure 5.11: Encoder silicon area

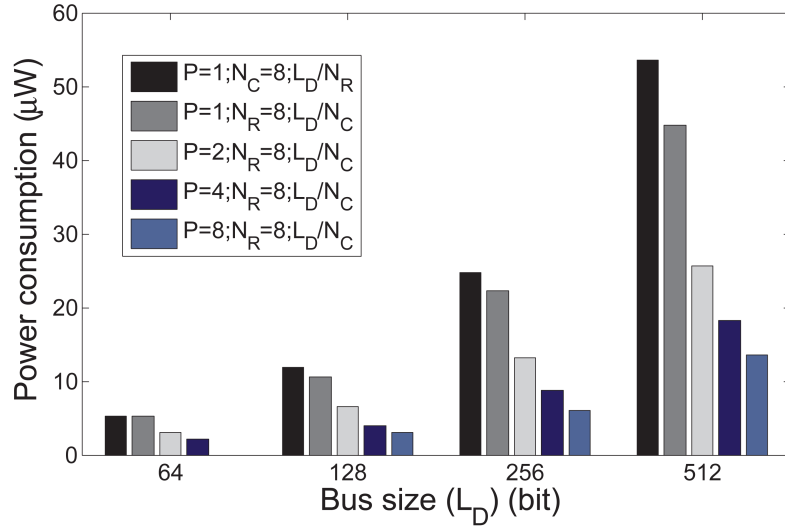


Figure 5.12: Proposed encoder power consumption versus TSV bus size

A TSV with 512 bits and 8 partitions occupies almost 20% of the area for the same bandwidth with one partition, as reported in Figure 5.11. For example, the power consumption of encoder in 512 bundle of TSVs with 8 partitions is almost 20% of the same bandwidth with one partition, as shown in Figure 5.12. Applications with more number of partitions have less power consumption due to hardware size reduction.

Table 5.3: Proposed encoder latency versus TSV bus size (*ps*)

Bus size	$N_C=8; N_R=L_D/N_C$	$N_R=8; N_C=L_D/N_R$			
	P=1	P=1	P=2	P=4	P=8
128-bit	144	207	147	112	97
256-bit	203	295	207	144	118
512-bit	639	283	219	221	151

## 5.6 Summary

Although 3D-NoC is a promising solution for exascale computing, its vulnerability to inductive TSV-to-TSV coupling has not been extensively studied. In this chapter, a coding algorithm is proposed to mitigate inductive TSV-to-TSV coupling after characterizing such issue by modifying input data stream. The ICM of the algorithm is then gauged in terms of various metrics such as mitigation measure, ICM gain, and data overhead. It is observed that the algorithm provides a significant ICM gain for relatively small bus sizes, while it suffers from a descending trend in performance as bus size increases.

With large bus size applications such as 3D-NoC, a partitioning approach is added to the algorithm to make it scalable with bus size. At the cost of reasonable overhead, significant ICM gain is obtained even in the case of large bus sizes. In addition to ICM gain and overhead, other practical issues such as power consumption and silicon area are also reported. It is observed that the coding approach promises lower power consumption and silicon area while providing considerable ICM gain for large bus sizes.

# Chapter 6

## Capacitive TSV Coupling Mitigation

In this chapter, two methods to mitigate TSV-to-TSV capacitive coupling are presented. These methods use information redundancy; basically, encoding/decoding algorithms are devised.

### 6.1 Introduction

The capacitance coupling between TSVs depends on the permittivity of the oxide, TSV geometry, the arrangement of surrounding TSVs, and body contacts places. TSV-to-TSV Capacitive Coupling (TTCC) is considered in this chapter as one of the major issues of 3D-IC design. I have previously proposed solely TSV-to-TSV inductive coupling aware coding for both low and high bandwidth application [28, 112], but they



are not proper for TTCC. Based on the analysis done in this thesis, the TSV configuration pattern which is the most interference-free in inductive and capacitive coupling analyses are different.

In this chapter, two TTCC mitigation coding methods for small and large 3D-IC bandwidths with affordable overhead are proposed. The proposed methods are scalable to support any  $n \times n$  number of TSVs without limiting any specific data patterns. The main contributions of this work are:

- To introduce the worst class of TTCC by a circuit-level analysis.
- To devise a baseline and an enhanced system-level method to mitigate the TTCC effect for smaller and larger bandwidths.
- To evaluate the efficiency and overhead of both proposed methods.

## 6.2 Proposed coding approaches

The main goal of the proposed TSV-to-TSV Capacitive Coupling Mitigation Algorithm (TCMA), is to reduce the probability of 7C and 8C parasitic capacitance patterns by adjusting the transmitting data bits. Mitigation is chosen since eliminating all 7C and 8C parasitic capacitance requires a complex coding approach which is not scalable for any size of TSV meshes [58]. In this section, the baseline TCMA is discussed for small interconnections and then issues of the baseline method for large interconnections are highlighted. Finally, the enhanced TCMA is presented which supports large mesh of TSVs.

### 6.2.1 Baseline TCMA

The TTCC is a data-dependent effect, as described earlier in Chapter 2. The basic idea of the baseline TCMA is to encode, if necessary, the consecutive data bits transmitting over the TSVs in order to mitigate the frequency of 7C and 8C parasitic capacitance. This method does not limit any pattern of data transmission bits by encoding them before transmission and decoding them in receiver side, if needed. The inversion operation is chosen as a simple but light and efficient practical coding method in TCMA in order to keep the overhead low, while mitigating TTCC noise. In a mesh of TSVs, a single bit per row is needed in TCMA to determine whether the inversion process is needed or not at the receiver side. TCMA stores the last transmitted data bit of each TSV and compares it with the available data bit which has not been transmitted yet. The current direction matrix of all TSVs is generated by comparing these successive data bits. Then the parasitic capacitance for each of TSVs are calculated based on the the current flow of its neighbor TSVs. Each row of 2D array of TSVs including 8C or 7C parasitic capacitance values is nominated for the data encoding process. By encoding the ready to transmit data bits, 8C parasitic capacitance will be converted to 4C, and 7C parasitic capacitance will be turned into 1C or 2C, in this method.

### 6.2.2 Enhanced TCMA

Although the baseline TCMA reduces the quantity of 8C and 7C parasitic capacitance values, but it may have some undesirable side effects by converting a row of data bits. For some special data patterns, converting a single row of data bits may unintentionally

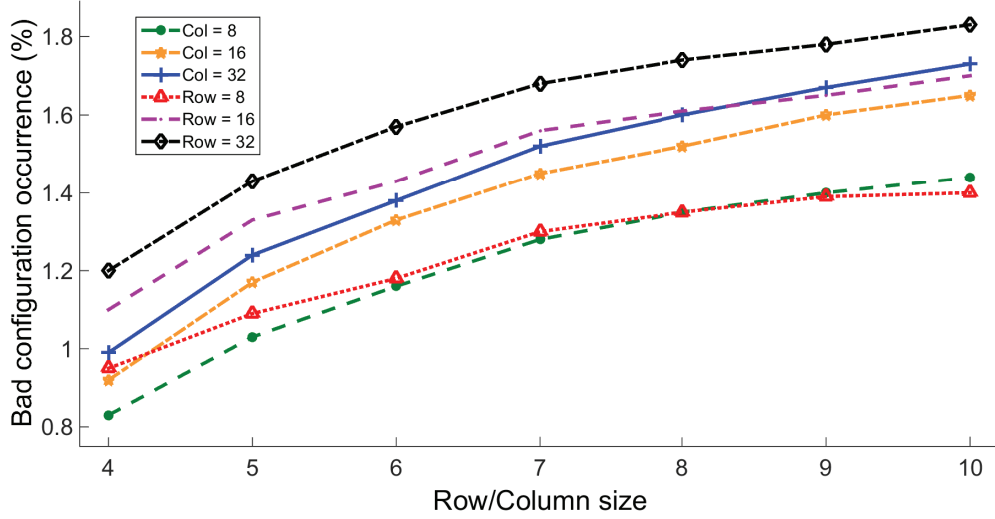


Figure 6.1: Probability of bad configuration occurrence

generate  $8C$  or  $7C$  parasitic capacitance values, which happens in a mesh of TSV with more than 3 rows or 6 columns. These special cases are referred as bad configuration in the rest of this chapter.

A bad configuration is a subset of TSV mesh which potentially generates unexpected  $8C$  or  $7C$  parasitic capacitance values by converting a single row of data bits. In more details, the row encoding affects the other data bits of the same row or the data bits in predecessor or successor rows in 2D matrix of TSVs. However, since the probability of bad configuration occurrence is low, specially for smaller matrices of TSVs, the baseline coding is still efficient for smaller data buses (less than 64 bits) which are considered in 3D-NoC. Figure 6.1 shows the probability of bad configuration occurrences in different mesh size of TSVs. This experiment is done by running Monte Carlo simulations for 10000 iterations for various row/column dimensions. According to experimental results, the reported percentage of bad configuration for all of the experimented dimensions is less than 2%.

Table 6.1: Current flow of TSVs before and after encoding

Sent data	Ready to send data	$CF_{bi}$	$CF_{ai}$
0	0	○	⊙
0	1	⊙	○
1	0	⊗	○
1	1	○	⊗

However, the baseline coding is not scalable for larger data buses (more than 64 bits) which are applied in 3D memory applications according to the increasing trend in Figure 6.1. The enhanced version of TCMA is devised for these sorts of application to make sure the encoding process of a selected TSV data bit does not worsen the total capacitive coupling.

Table 6.1 summarizes the TSV current flow direction before and after encoding its ready to send data bit.  $CF_{bi}$  shows the current flow of TSV before inverting the ready to send data, while  $CF_{ai}$  represents the current flow of TSV after inversion. Based on this table, an inactive TSV current flow (○) may convert to active TSV (either ⊙ or ⊗), while an active TSV (either of the ⊙ or ⊗) is converted into an inactive one (○) after inverting the ready to send data bits. Based on the analysis, a bad configuration occurs in five cases, while two of them are potential to generate unwanted 8C parasitic capacitance and the other three may generate unwanted 7C parasitic capacitance. They are called `bad_config8,1`, `bad_config8,2`, `bad_config7,1`, `bad_config7,2`, and `bad_config7,3`. Figure 6.2 illustrates these five cases in top view of 2D array of TSVs in a 3x3 mesh of TSVs. The candidate row for inversion is recognized by dashed lines in this figure. It also shows the parasitic capacitance value of middle TSV in the recognized row by dashed lines before and after encoding process. Each of these bad configurations affects the result of baseline coding with some conditions which are

discussed in the next.

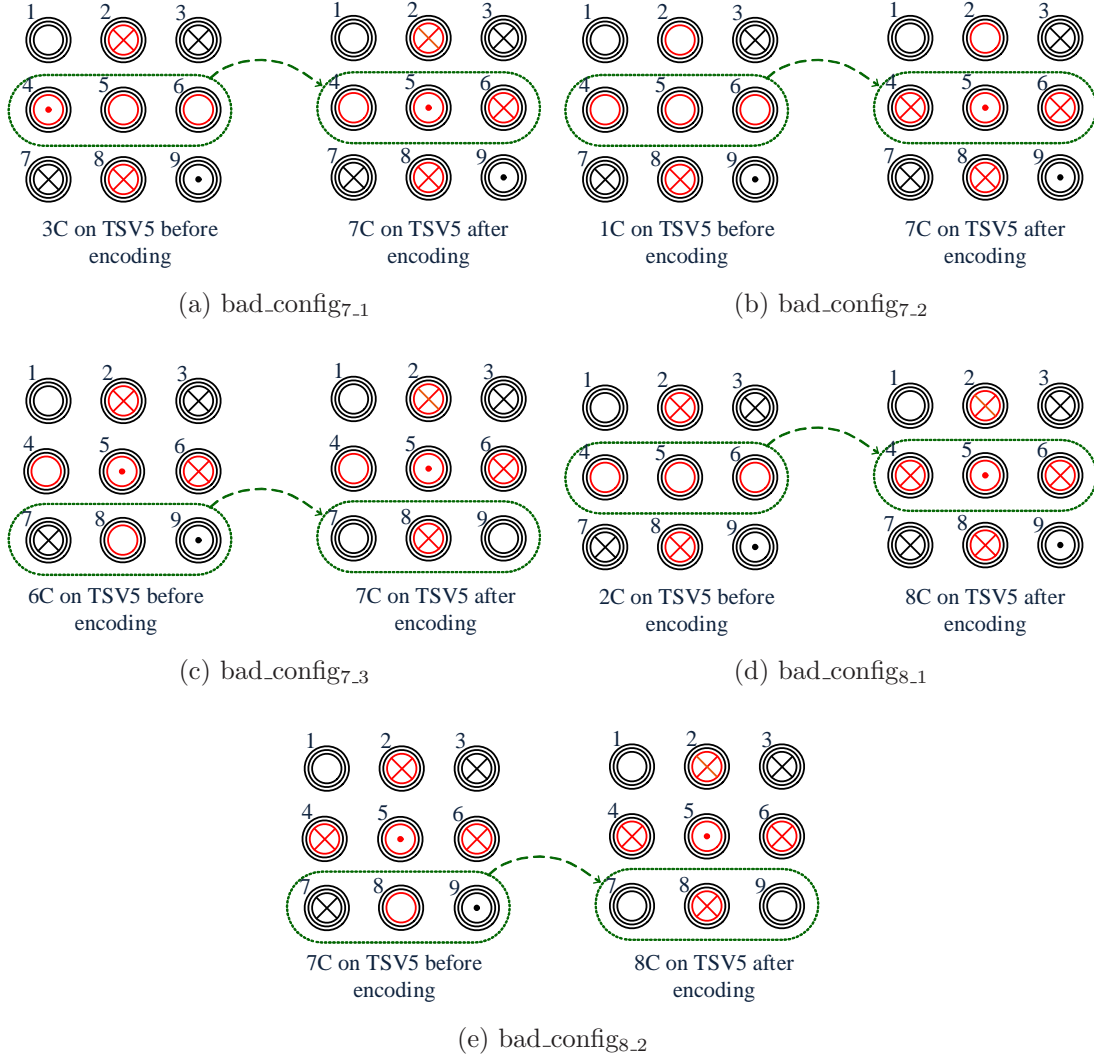


Figure 6.2: Potential configurations to generate 7C and 8C parasitic capacitance

In the baseline method and in case of encoding, the 3C parasitic capacitance, if any, is converted into 7C (see Figure 6.2a) by encoding the second row of 2D array of TSVs with following four conditions:

- There are exactly two inactive TSV next to each other in potential row for

encoding process as in TSV5 and TSV6.

- TSV2 and TSV8 are active with the same current direction.
- The current direction of TSV6 after encoding should be the same as the current direction of TSV2 and TSV8.
- The current direction of TSV5 should be reverse of the current direction in TSV2, TSV6, and TSV8 after encoding.

The 1C parasitic capacitance is converted into 7C (see Figure 6.2b) by encoding the second row of 2D array of TSVs with following four conditions:

- There are at least three inactive TSV next to each other in potential row for encoding process as in TSV4, TSV5, and TSV6.
- Either of TSV2 or TSV8 is inactive and the other should be active.
- The current direction of TSV4 and TSV6 after encoding should be the same as the current direction of either TSV2 or TSV8 which was active.
- The current direction of TSV5 after encoding should be reverse of the current direction of TSV4, TSV6, and either TSV2 or TSV8 which was active.

The 6C parasitic capacitance is converted into 7C (see Figure 6.2c) by encoding the third row of the 2D array of TSVs with following four conditions:

- In capacitive matrix there is a 6C parasitic capacitance in preceding row which is selected for encoding in a way that TSV5 has reverse current direction of TSV2 and either of TSV4 or TSV6.

- TSV8 which is in the nominated row for encoding is inactive.
- One of TSV4 or TSV6 is inactive and the other should be active with reverse current direction of TSV5.
- The current direction of TSV8 after encoding should be same as current direction of TSV2 and either of TSV4 or TSV6 which was active.

The 2C parasitic capacitance is converted into 8C (see Figure 6.2d) by encoding the second row of 2D array of TSVs with following four conditions:

- There are at least three inactive TSVs beside each other in potential row for encoding process like TSV4, TSV5, and TSV6.
- TSV2 and TSV8 are active with same current direction.
- The current direction of TSV4 and TSV6 after encoding should be the same as the current direction of TSV2 and TSV8.
- The current direction of TSV5 should be reverse of the current direction in TSV2, TSV4, TSV6, and TSV8 after encoding.

The 7C parasitic capacitance is converted into 8C (see Figure 6.2e) by encoding the third row of 2D array TSV, if the following conditions are satisfied:

- In capacitive matrix there is a 7C parasitic capacitance in predecessor row which is selected for encoding. The inactive TSV should be also in the selected row for encoding.

---

**Algorithm 6.1.** Enhanced TCMA coding algorithm

---

```
1: AMAT  $\leftarrow$  Sent data bits
2: BMAT  $\leftarrow$  To be sent data bits
3: CMAT  $\leftarrow$  Current direction of each TSV generated by AMAT & BMAT
4: CAPMAT  $\leftarrow$  Capacitive parasitic noise of each TSV generated by CMAT
5: INV  $\leftarrow$  Redundant vector for inversion process decision at receiver side
6: for each  $R \in Rows$  do
7:   for each  $C \in Columns$  do
8:     if CAPMAT[R][C] == 8 or CAPMAT[R][C] == 7 then
9:        $78C\_counter ++$ 
10:    end if
11:    if (there is a bad configuration bad_config7.1 or bad_config7.2 or
12:    bad_config7.3) then
13:       $bad\_config_7\_counter ++$ 
14:    end if
15:    if (there is a bad_config8.1 or bad_config8.2) then
16:       $bad\_config_8\_counter ++$ 
17:    end if
18:    end for
19:    if ( $78C\_counter > bad\_config_7\_counter + bad\_config_8\_counter$ ) then
20:      Encode the BMAT[R]
21:      INV[R]=1
22:    end if
23:  end for
```

---

- TSV8 has the reverse current direction of TSV5 after encoding process.

The probability of bad configuration presence in a mesh of TSVs is very low since all the discussed conditions should be satisfied simultaneously. However, the goal of the enhanced TCMA, which is summarized in Algorithm 6.1, is to guarantee the encoding process will not worsen the total number of 7C and 8C parasitic capacitance in a 2D array of TSVs. In the enhanced version of TCMA, the encoding process will be done if the total number of 7C and 8C parasitic capacitance in capacitive matrix is higher than the total number of bad configuration in each row.



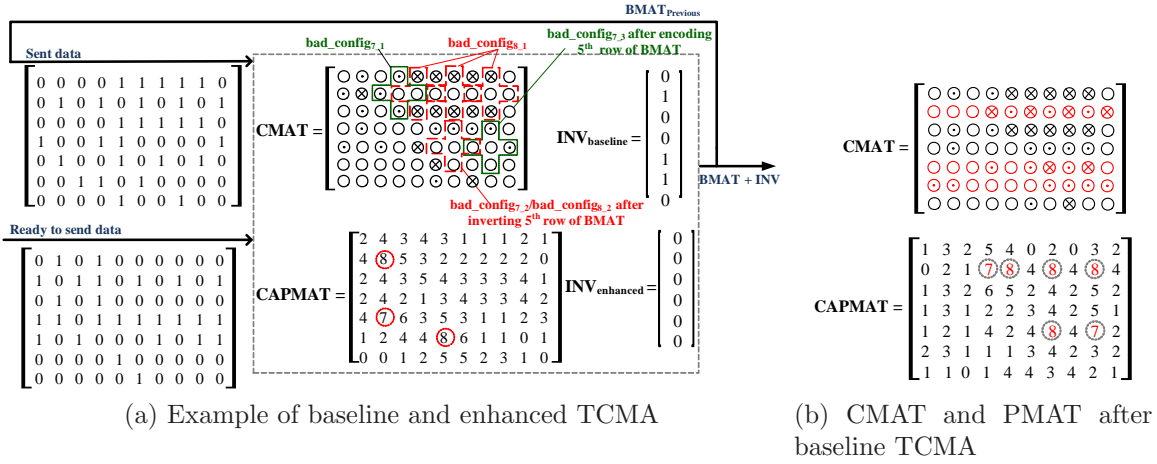


Figure 6.3: An example of baseline algorithm issues

### 6.3 TCMA Elaboration and Evaluation

Figure 6.3a illustrates an example of the baseline and enhanced algorithms for  $7 \times 10$  given AMAT and BMAT matrices. These matrices and the ones which are used in following sentences are defined in Algorithm 6.1. This dimension has been chosen to show the advantages of the enhanced approach over the baseline technique for higher bandwidth data buses. First, the CMAT and then the CAPMAT matrices are generated from the transmitted (AMAT) and to be transmitted (BMAT) data lines. Then, CAPMAT is generated from CMAT by counting the total mutual capacitive parasitic difference between each TSV and its adjacent neighbors. The INV matrix is evaluated in the receiver side to extract the original data values if they are encoded.  $INV_{baseline}$  of this example shows that the second, fifth, and sixth rows of the BMAT matrix have been encoded since there are 8C or 7C parasitic capacitance values in these rows of CAPMAT matrix. Since the number of 7C and 8C parasitic capacitance are not higher than the number of bad configuration in enhanced method, the  $INV_{enhanced}$  shows none

of the rows the BMAT has been encoded. Figure 6.3b represents the updated CMAT and CAPMAT matrices in the baseline approach after encoding the second, fifth, and sixth rows of BMAT matrix in which the total number of 7C and 8C parasitic capacitance increases from 3 to 6. This example illustrates all 5 possible bad configurations. The `bad_config7,1` and `bad_config8,1` are depicted in second row of CMAT in Figure 6.3a, resulting in three 8C and one 7C after encoding second row of BMAT. The `bad_config7,2` of fifth row is highlighted in CMAT matrix of Figure 6.3a. After encoding fifth row of BMAT, the undesirable 7C will be generated in CAPMAT[5][7], which is also `bad_config8,2`. Furthermore, encoding fifth row of BMAT generates a `bad_config7,3` in CAMPMAT[5][9]. Since the encoding decision is supposed to be made row by row in one direction (from top to bottom in this example) or reverse, the unwanted generated 7C and 6C in fifth row of CAPMAT are potential to generate 8C and 7C, respectively by encoding the sixth row of BMAT. Due to the presence of 8C in sixth row of CAPMAT, it is selected for encoding process and both of `bad_config7,3` and `bad_config8,2` generate undesirable 8C and 7C in fifth row of CAPMAT which is shown in Figure 6.3b. However, the enhanced algorithm prevents these bad effects by predicting them.

To evaluate the advantages of the baseline TCMA for smaller mesh size, Monte Carlo simulations for 10000 iterations on different sizes of TSV mesh are examined. The total number of 7C and 8C parasitic capacitance before and after applying the baseline TCMA for different mesh size of TSVs is shown in Figure 6.4. It is depicted that the mitigation rate of 7C and 8C parasitic capacitance after applying the baseline TCMA are almost 98%, 94%, and 90% for  $4 \times 4$ ,  $6 \times 6$ , and  $8 \times 8$  TSV meshes. The information redundancy of the baseline TCMA method for these sizes of mesh of

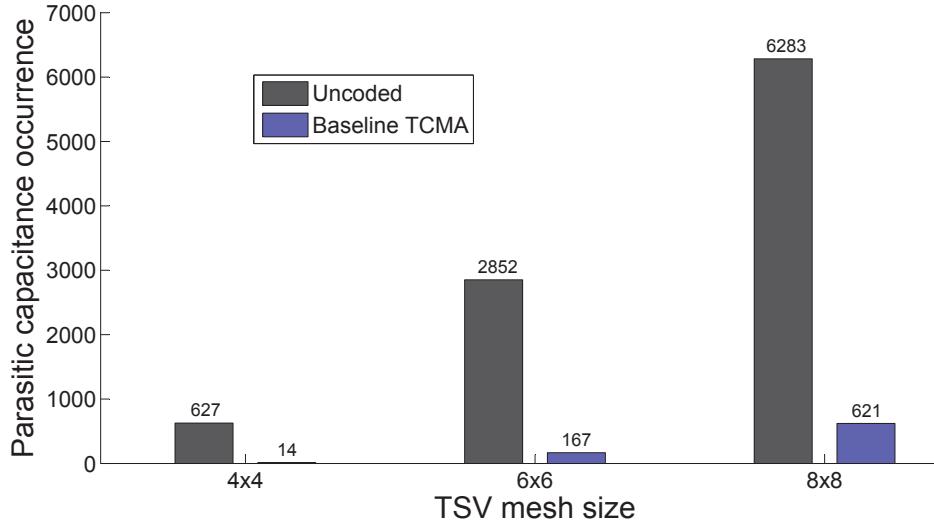
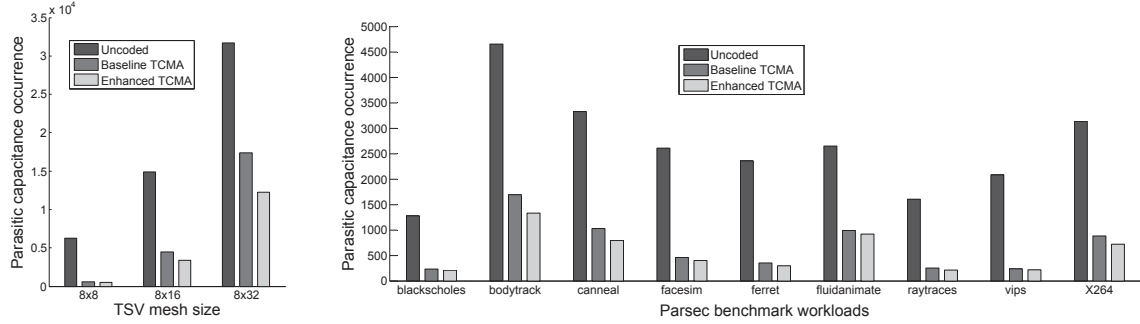


Figure 6.4: Number of 7C/8C for random data bit patterns in small mesh of TSVs

TSVs are 25%, 16%, and 12%. However, the mitigation rate of the baseline TCMA is increased for large mesh of TSVs, as expected. This is because of the probability of bad configuration occurrence rises by increasing the sizes of TSV meshes. The Monte Carlo simulations for 10000 iterations for larger mesh of TSVs are also examined for both baseline and enhanced TCMA to show the advantages of enhanced TCMA. Although the mitigation rate of total number of 7C and 8C parasitic capacitance values is increasing by using larger mesh of TSVs, enhanced TCMA prevents encoding process if the result is worsen. This is shown in Figure 6.5a, in which the mitigation rate of 7C and 8C parasitic capacitance occurrence by applying enhanced TCMA are always higher than baseline approach. PARSEC benchmark [11] as a realistic data traffic for large size of mesh of TSVs are also applied to check the performance of the baseline and enhanced TCMA. Memory traces of PARSEC applications have been employed in this experiment, which are extracted by the PIN tool [39], a dynamic binary instrumentation framework for the IA-32 and x86-64 instruction-set architectures. The total number of 7C and 8C parasitic capacitance values for memory traces of PARSEC



(a) Random data bit patterns in larger mesh of TSVs

(b) PARSEC application data bit patterns in  $8 \times 32$  mesh of TSVs

Figure 6.5: 7C and 8C parasitic capacitance for random and PARSEC applications data with/without TCMA

Table 6.2: Hardware synthesise results

Mesh size	Baseline		Enhanced	
	Area ( $\mu m^2$ )	Power ( $\mu W$ )	Area ( $\mu m^2$ )	Power ( $\mu W$ )
$8 \times 8$	918	2340	1096	3000
$8 \times 16$	1818	4520	2173	5900
$16 \times 8$	2094	5260	2165	5880
$8 \times 32$	3321	8840	4331	11700
$32 \times 8$	4086	11000	4323	11700

application workloads through the TSVs are reported for a  $8 \times 32$  mesh of TSVs in Figure 6.5b. The migration rate of TCMA for blackscholes, facesim, vips, and raytraces are between 80% to 90%, and for the rest of them is almost 70%. Although the differences between the mitigation rates of baseline and enhanced TCMA are not very much, but the result of enhanced method is always better than baseline, as it is expected. In other words, it is always guaranteed that by applying the enhanced TCMA the total number of 7C and 8C parasitic capacitance will never be worse off because of the bad configuration presence.

In order to evaluate the proposed coding methods, the baseline and enhanced

TCMA encoders are implemented in Verilog and synthesized by Synopsys Design Compiler using 28nm TSMC library (1.05V, 25 °C). Table 6.2 reports the synthesis results for power consumption and occupied area. The latency of the enhanced method is reported by the critical path including: registers latching the adjusted output data bits toward the feedback input for subsequent CMAT computation. In other words, it does not depend on the dimension of TSV arrays. According to the logic synthesis, the latency of the baseline and enhanced TCMA are reported as 69.5ps and 74.9ps for all given TSV dimensions in Table 6.2. The feasibility of both proposed coding algorithms are confirmed by considering the gained coupled parasitic capacitance mitigation and its tangible footprint and power consumption. Decoder units are not implemented in this experiment since they are only composed of a comparator and a mix of inverter gates. They are much lighter than encoder components in terms of area, power consumption, and latency.

## 6.4 Summary

Two baseline and enhanced algorithms have been proposed in order to minimize the TSV-to-TSV capacitive coupling issue. Baseline algorithm is proposed for small mesh of TSVs which are considered in 3D NoC applications, while enhanced method is suggested for large mesh of TSVs which are more applied in 3D memory applications. The enhanced method guarantees that the encoding process prevents generating undesirable parasitic capacitance values by recognizing all susceptible configurations. According to experimental results, the baseline method's mitigation rate is more than 90% for TSV meshes smaller than  $10 \times 10$ . The enhanced algorithm mitigates the

TSV-to-TSV capacitive coupling more than 70% for  $8 \times 32$  mesh of TSVs.

# Chapter 7

## Asynchronous Architecture to Avoid TSV Coupling

### 7.1 Introduction

Many research groups have proposed robust techniques for mitigating TSV-to-TSV coupling in order to increase the reliability of 3D multiple-stacked ICs. In order to block the coupling path between data-transmission TSV, grounded TSVs can be added either surrounding the victim TSV [62] or into the empty spots of the design [98]. However, these approaches impose large area overhead and design complexity, since the grounded TSVs are larger than data transmission TSVs [84]. Employing other types of grounded barriers have also been proposed to minimize area overhead while reducing the coupling. Utilizing guard rings around the victim TSV to form a stronger discharging path and adopting differential signaling have also been proposed in order to

improve immunity of design against TSV-to-TSV coupling [84]. However, these methods limit the TSV placement and arrangement in the fabrication process. A coding scheme has been suggested for a matrix of TSVs, reducing the maximum crosstalk by 25% [58]. However, this approach is only applicable for capacitive coupling and only supports a mesh of TSVs with size of  $3 \times n$ , limiting the TSV insertion process. In addition, it imposes around 40% information redundancy with an encoder and decoder of quadratic circuit complexity.

The goal of this chapter is to design an architecture for Capacitive TSV-to-TSV Coupling Avoidance (CTCA). In this chapter, first, a scalable approach is devised that eliminates the worst undesirable parasitic capacitive cases induced by CTTC (larger than  $4C$ ), detailed in Section 7.2. Subsequently, the efficiency of the proposed approach is demonstrated and justify their practicality through concrete experiments, both in hardware and system levels, illustrated in Section 7.3.

## 7.2 CTCA Technique

The main goal of the proposed CTCA approach is to eliminate all the parasitic capacitive noises larger than  $4C$ . Eliminating all types of parasitic capacitive noises impose a complex architecture which is not scalable to applications that need larger number of TSVs [58, 29]. However, in this chapter, a fast architecture is proposed which implements the dual-rail coding without a tangible overhead. The dual-rail coding is first explained in this section. Then, the proposed coding architecture is presented and finally, the characteristics of the proposed architecture in removing parasitic capacitance noises larger than  $4C$  is discussed.



## 7.2.1 Dual-rail Coding Concept

Dual-rail coding is composed of two different states of data transmission: neutral and valid states. For the valid state, one of the rails is asserted to transmit a '0,' while the other rail is asserted to transmit a '1' value. The asserted rail is then reset to '0' before the next data value is transmitted, which is called neutral state. In other words, in this approach data transmission is accomplished through a valid state which is followed by a neutral state, as shown in Figure 7.1. In this method transferring "11" symbol through the rails is prohibited [74].

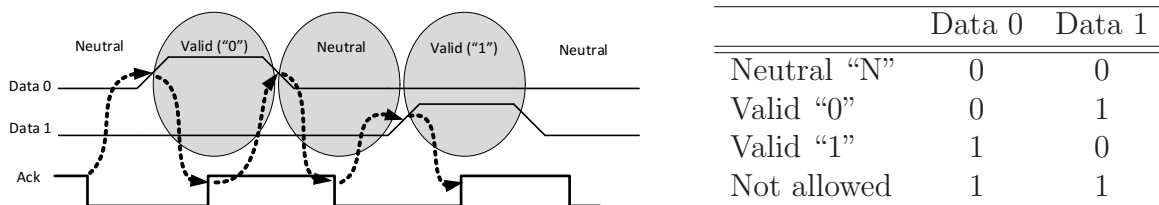


Figure 7.1: Dual-rail encoding

Four-phased communication protocol is combined with dual-rail coding method to implement proposed asynchronous communication of the proposed CTCA approach. For the four-phased protocol, a receive process consists of the following four steps:

1. Wait for the input to become valid.
2. Acknowledge to the sender that the transmission has been accomplished.
3. Wait for the inputs to become neutral.
4. Make the acknowledge signal low.

On the other side a send activity includes four subsequent phases:

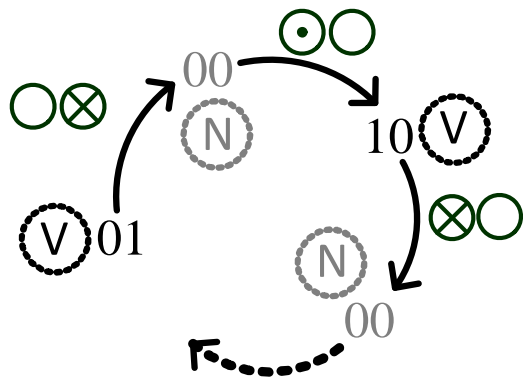
1. Send a valid output.
2. Wait for acknowledge.
3. Make the output neutral.
4. Wait for acknowledge to lower output.

A dual-rail encoding has been employed in such implementation in which the data channel contains a valid data (token) when just one of two TSV is transferring data value of '1'. When the both TSVs are transferring data value '0,' the channel contains no valid data and is called to be in the neutral state. The advantages of applying dual-rail coding as a CTCA method is discussed in following subsection.

## **7.2.2 Parasitic capacitance elimination with proposed coding**

In dual-rail coding approach, half of the TSVs are always inactive while the other half are all active with the same current flow direction. This occurs because the source of all data TSVs are ready to transfer at the same time and also data-rails are in neutral state between each sequence of data valid states. In the neutral state, all TSVs are supposed to transfer logic '0' value from the sender towards the receiver. Furthermore, in the valid state the sender value for each TSV changes from '0' to '1' or it remains '0' depending on the value of data bit on the sender side.

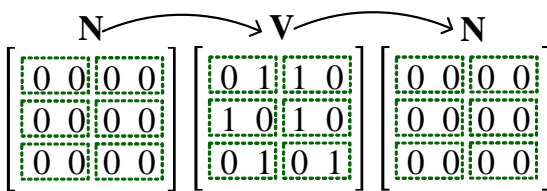
Figure 7.2 illustrates TSVs' current flow in dual-rail coding approach in general and for a given example with 6-bit bandwidth. Assume there is a data transmission through dual-rail TSVs from upper to lower level. With this assumption, half of the



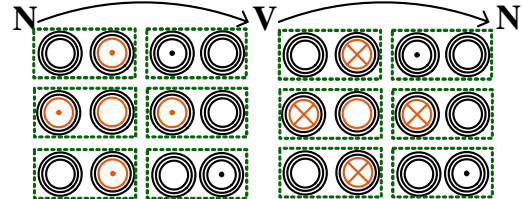
(a) TSVs' current flow in transition between valid and neutral states

Original	Coded
$\begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$

(b) Original and coded data for a given example of 6-bit data bandwidth



(c) Ready to transfer data bits through TSVs in neutral and valid states for the given example



(d) TSVs' current flow after transition from neutral to valid and valid to neutral states in given example

Figure 7.2: TSVs' current flow in dual-rail coding

TSVs' current flow directions are  $\circ$  and the other half are  $\otimes$  in transition from a valid to neutral state, while in transition from neutral to valid state half of all TSVs' currents are  $\circ$  while the other half are  $\odot$ , as shown in Figure 7.2. In this figure,  $N$  stands for neutral state and  $V$  represents the valid state. The TSVs' current flow is reverse in transitions from neutral to valid and from valid to neutral states, if the location of sender and receiver are switched.

Figure 7.2b represents the original and their corresponding dual-rail coded values for a given example with 6-bit bandwidth. As it is shown in Section 7.2.3, 18 TSVs are needed to transfer 6 data bits in dual-rail approach with the four-phased protocol (12 for data transmission and 6 for handshaking). In this method it is assumed that all

handshaking TSVs are separated from data transferring TSVs. There is no capacitive coupling effect among TSVs for transferring data bits and handshaking signals. Figure 7.2c shows the corresponding 12 data bits for the given 6 bits. Each pair of data bit patterns separated by dashed lines presents an encoded single data bit. As discussed earlier, each pair of TSVs are supposed to transfer “00” values in the neutral state, and “01” or “10” are transferred through TSVs depends on the value in the source side of TSV (see Figure 7.1). Figure 7.2d represents the current direction of 12 TSVs after transition from neutral to valid and from valid to neutral states. The presented example, depicted in Figure 7.2, is the worst possible data bit pattern in which 4C parasitic capacitance occurs. Similarly, 4C parasitic capacitance is generated for transition from neutral state to valid state in the worst case. With this protocol that the current flow of no TSV changes from  $\otimes$  to  $\odot$  or reverse, the elimination of parasitic capacitance values larger than 4C is guaranteed. Furthermore, since all data transmissions are supposed to be done at the same time, the handshaking TSVs are always asserted to the same data bit values. Effectively, they are all supposed to transmit ‘1’ or ‘0’ values from the same sender side simultaneously. Based on this fact, handshaking TSV’s current flow are always similar; no undesirable parasitic capacitance signal is generated over handshaking TSVs.

### 7.2.3 Supporting Architecture for Dual-rail Coding

The maximum number of TSVs used in 3D-IC designs is limited due to their large dimensions as compared to the die size [92, 124]. In order to meet this restriction, the proposed CTCA approach does not require applying extra TSVs, while it eliminates the parasitic capacitance noises larger than 4C. This restriction is met despite of the

fact that three TSVs are required in dual-rail coding to transfer one data bit. As discussed in Subsection 7.2.1, each data bit is transmitted using two links and one additional link is utilized for the handshaking process. Therefore, in order to maintain same communication throughput, while presenting a capacitive coupling tolerant approach, data transmission through TSVs should be done at least three times faster. In other words, in each clock cycle at least three data bits should be transferred through the TSV lines to compensate applying three TSVs per bit in dual-rail coding.

In this thesis, a fast asynchronous link architecture is proposed in order to implement the dual-rail coding with four-phased protocol which is capable of transferring four bits per clock cycle. Figure 7.3 illustrates the proposed architecture, in which four serializer and deserializer are needed to transfer 4 bits in each clock cycle. Two Globally Synchronous Locally Asynchronous (GALS) wrappers are also required to handle communication between synchronous and asynchronous islands. In more detail, the serializer is derived by *Sync2Async* modules and the deserializer is connected to *Async2Sync* modules, as shown in Figure 7.3. Also a self-controlled SerDes [24] is employed in this design between a Sync2Asny and Async2Sync pairs, since they do not require the clock signal.

These SerDes components are responsible to transmit four data bits at each clock cycle. The circuit of two GALS wrappers [114], are also shown in Figure 7.4. The dual-rail encoding and decoding process and synchronization are done in these wrappers. The State Transition Graph (STG) of the proposed synchronous to asynchronous wrapper is shown in Figure 7.5a. The synchronous to asynchronous interface waits for the *Request* signal to rise, driven by synchronous module. The *Enable* signal is activated once the *Request* signal rises while value of the *InputAck* signal is low.

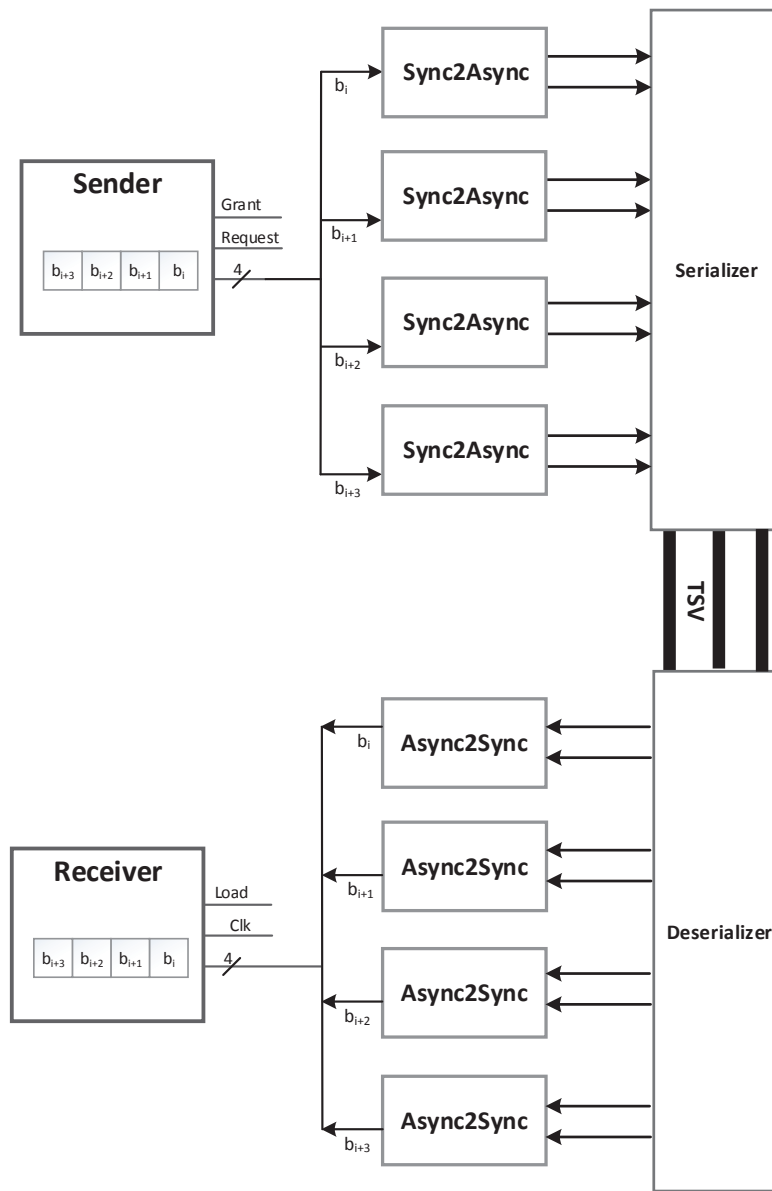


Figure 7.3: Dual-rail coding implemented using GALS architecture

*Enable* signal activation, permits the transmission of synchronous data through the asynchronous module. The *InputAck* signal of asynchronous side and *Grant* signal in synchronous module rise as the incoming data from synchronous unit encodes into dual-rail mode. *Grant* signal activation in synchronous module results in inactivating

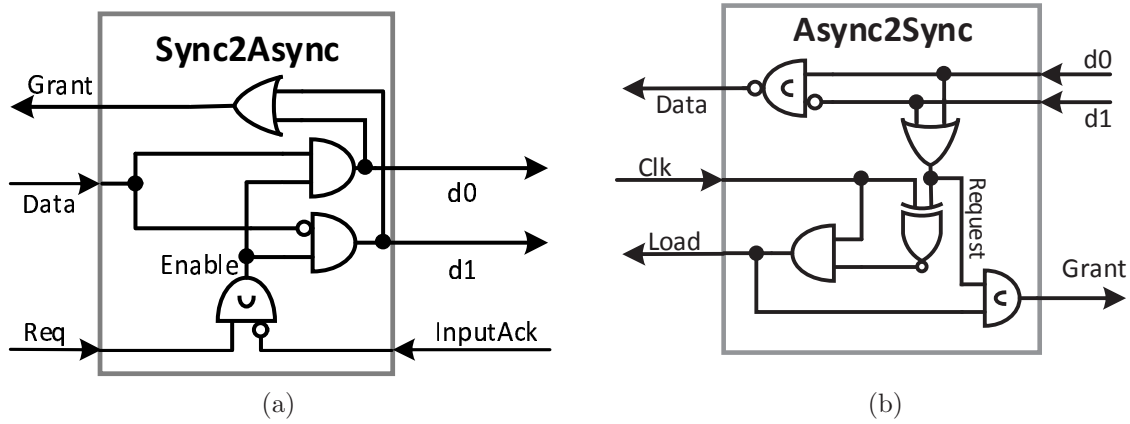


Figure 7.4: GALS wrapper circuit, (a) sync to async, (b) async to sync

the *Request* and consequently *Enable* signal and switching from valid state into neutral state. The data path waits for the succeeding data by inactivating the *Grant* signal.

Similarly, the related STG of asynchronous to synchronous design is shown in Figure 7.5b. In this protocol, first the *Request* signal rises to announce the transmission demand towards the synchronous module, once the asynchronous data output lines,  $d_0$  and  $d_1$ , are valid. At synchronous side, the *Request* signal is examined at each rising edge of the clock signal (CLK). The *Load* pin of the D flip-flap is activated, if the synchronous module detects the *Request* signal at rising edge of the clock. The asynchronous sender block is aware of reading its generated data when it detects both the *Request* and *Grant* signals as high. Such a situation causes the asynchronous side to assert neutral data through the data lines. The *Request* signal falls down once the neutral data appears on the data lines, which results in inactivating the *Load* signal in the subsequent falling edge of the clock.

The only constraint for this architecture is that the total delay of the proposed architecture and TSV delay for transmitting  $N$  data bits should be at least  $N$  times

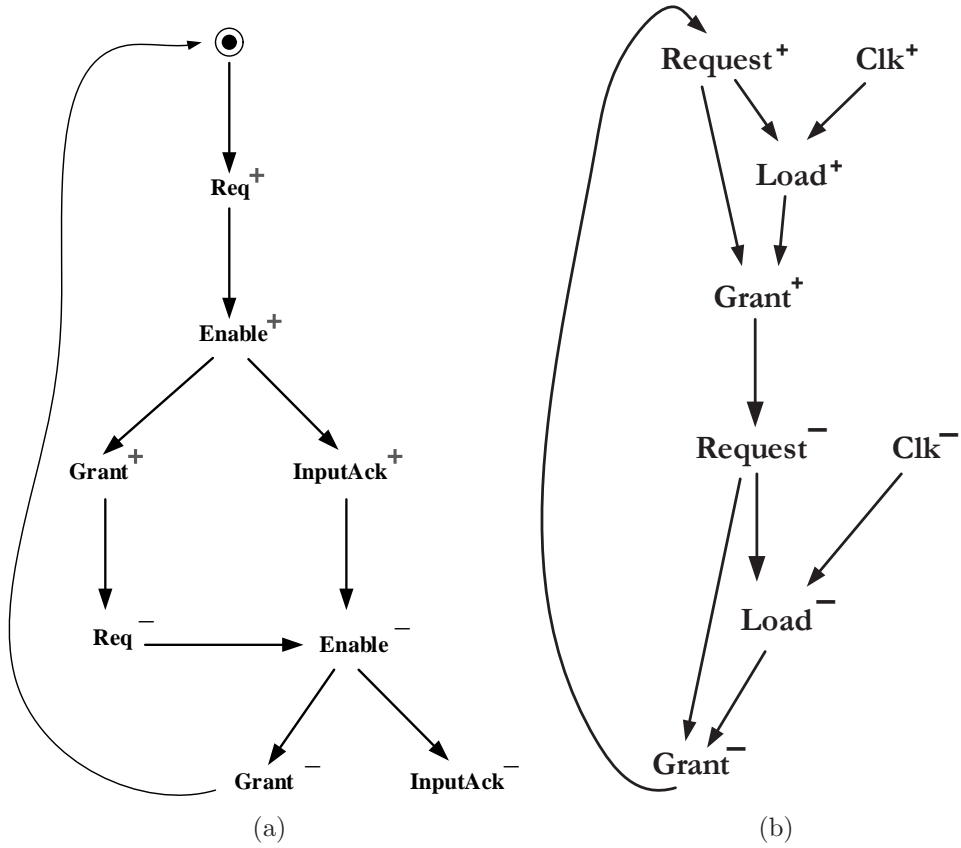


Figure 7.5: GALS wrapper STGs, (a) sync to async, (b) async to sync

less than the systems’s clock cycle period. The analysis, reported in Section 7.3, proves that the constraint is met in the proposed architecture. Considering latency of SerDes, GALS wrappers, and TSV link, the proposed architecture runs at 800MHz frequency.

### 7.3 Evaluation of CTCA Methods

To validate the efficacy of the proposed CTCA method, it has been simulated for an  $8 \times 8$  mesh of TSVs. In this experiment, TSVs are derived by a random and realistic data traffic, in which the occurrence frequency of parasitic capacitance factors before



Table 7.1: Circuit-level model results of CTCA

	<b>Area</b>	<b>Latency</b>	<b>Power</b>
4× Sync2Async	1.8 $\mu m^2$	29ps	23 $\mu W$
Serializer <sub>4×1</sub>	5.03 $\mu m^2$	32ps	45 $\mu W$
3×TSV drivers	2.28 $\mu m^2$	29ps	90 $\mu W$
Deserializer <sub>1×4</sub>	4.74 $\mu m^2$	31ps	40 $\mu W$
4× Async2Sync	4.53 $\mu m^2$	18ps	45 $\mu W$
Total	18.4 $\mu m^2$	139ps	243 $\mu W$

and after applying CTCA method are compared. PARSEC [12] benchmark memory traces have been collected using PIN [39] as a real-world data traffic. The occurrence frequency parasitic capacitance values are shown for uncoded (left bar) and CTCA approach (right bar) for both random (see Figure 7.6a) and PARSEC benchmark (see Figure 7.6b) data traffic.

It is observed that CTCA completely eliminates the adverse parasitic capacitance values larger than 4C (5C, 6C, 7C, and 8C); instead, as a result, these parasitic capacitance classes are converted to 1C and 2C cases.

The colors in Figure 7.6 are selected in a way to emphasize the adversity of each case. In addition, 0C, 1C, and 2C cases are not shown in this figure. Based on results in [58], capacitive cases 6C, 7C, and 8C, depending on the timing requirement, are unsafe and most likely to result in failure condition. As it is expected, in all applications, unsafe cases are avoided in CTCA (right bars), and the occurrence frequency of safe cases, 3C and 4C, are increased, instead.

The other benefit of the proposed CTCA approach is its scalability property due to employing asynchronous architecture. It means that by increasing the number of TSVs in rows or columns, the capability of the proposed CTCA is not affected without

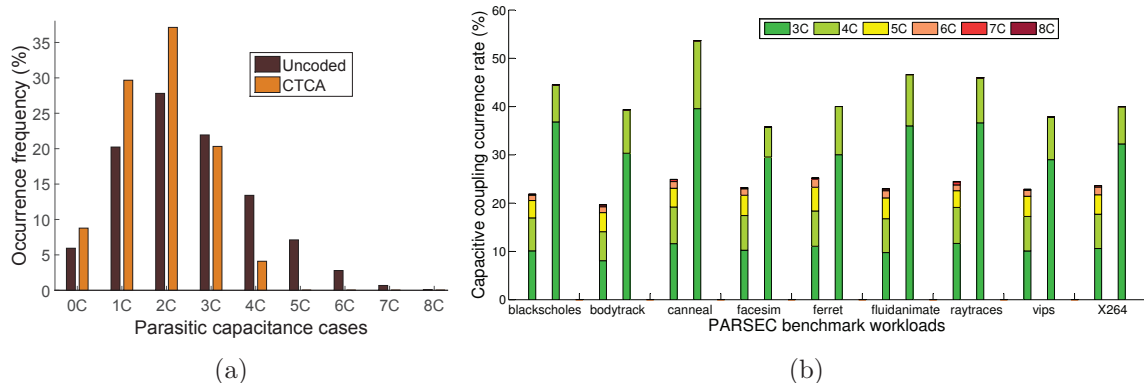


Figure 7.6: Capacitive coupling percentages before and after employing CTCA for both random and PARSEC workloads

exposing unexpected overhead. The reason is that the parasitic capacitance elimination in CTCA is handled independent of current flow in adjacent TSV pairs. Therefore, increasing the number of TSVs does not impact the behavior of dual-rail coding nor the asynchronous architecture sequence in CTCA approach; hence the parasitic capacitance elimination rate is not changed by increasing the dimensions of TSV arrays.

In order to validate the functionality of the proposed CTCA approach, the circuit-level model, including the SerDes, GALS wrappers, and TSVs, is developed. The circuit-level model is then simulated with 22nm PTM library using Synopsys HSPICE tool. The corresponding simulation results are reported in Table 7.1. It should be noted that these hardware results are only for four bits data transfer. In order to estimate the total area and power consumption for  $N$  bit bus design, corresponding results can be easily multiplied by  $N/4$ . The proposed circuit for using dual-rail coding imposes a negligible amount of area and power. Also, comparing to the clock cycle period of synchronous data sender block, such as NoC routers which usually work at 1GHz [115, 80], with total latency of proposed asynchronous transmitter block which

has 139ps (over 6GHz in total for one transmission), confirms that the proposed block works effectively fast. It should be also noted the other advantage of the proposed CTCA method is to reduce 25% of area occupied by TSVs, since it is capable of transmitting each 4 bits with 3 TSVs.

## 7.4 Summary

Although 3D-IC architecture is a promising approach for exascale computing, its vulnerability to TSV-to-TSV coupling needs to be entirely addressed. Having characterized the TSV coupling, in this chapter, a coding approach along with its architecture design is proposed to mitigate capacitive aspect of TSV-to-TSV coupling effect. To heal the capacitive TSV coupling issue, It is proposed to use dual-rail coding for transferring data. This method helps to eliminate the critical capacitive coupling cases. Subsequently, a supporting GALS architecture is designed to implement dual-rail coding without imposing any TSV overhead. Later, the proposed coding method is modeled and verified at system-level. The performance is measured using realistic and random traffic. Also, the method is implemented at hardware-level and the corresponding results are reported.

# Chapter 8

## Conclusions and Future Roadmap

### 8.1 Put it All Together

3D-IC designs are considered to be a viable solution for integrating more cores on a chip, while imposing a smaller footprint area and better timing performance as compared to 2D architectures. In a 3D package, multiple dies are stacked together with direct vertical interconnects. There are various vertical interconnect technologies. Wire-bonding and flip-chip stacking methods were first proposed to implement 3D-IC designs, but TSV is currently more attractive as a vertical-electrical connection form between tiers. TSV-based 3D IC design techniques support better performance and integrated functionality by shortening both average and maximum distances among components on different dies

Although, 3D-IC architecture is a promising approach for exascale computing, its resiliency is a factor which may hinder it. Reliability is one of the most challenging

problems in the context of 3D-NoC systems, as an application for 3D-IC integration. Highly accurate and efficient reliability analyses, fault models, and fault-avoidance and fault-tolerant methods are required to enhance the reliability of this HPC systems. Reliability analysis is prominent for early stages of the manufacturing process in order to prevent costly redesigns of such an expensive 3D system. Fault models specific to 3D technology help in better analyzing the reliability of 3D architectures at run-time. Using fault-avoidance techniques, probability of failure and hence the power consumption of error detection mechanism would be lower. And finally, developing fault-tolerant mechanisms which are uniquely designed for 3D systems will highly improve the reliability of the system while maintaining the performance and power.

## 8.2 Summary of Contributions

I summarize the contributions of this thesis in the following sections.

- **Comprehensive study and analysis of reliability issues in 3D designs**

In Chapter 2, potential sources of physical faults in 3D-NoC are highlighted and their corresponding logic-level fault models are presented. Also, the impacts of all potential physical faults on 3D-NoC components are addressed. It provides reliability metrics for 3D-NoC environment.

Chapter 3 modeled TSV characteristics as a time-invariant failure probability. In this chapter, a reliability criterion for TSV-based NoC is defined as the probability of having at least one faulty TSV in a given time slot. Consequently, the relationship between NoC reliability and TSV failure is quantified, and the

resulting equation is reduced to a tractable form with manageable computational complexity. The final equation relating the NoC reliability criterion and TSV failure includes a non-analytical probabilistic term which can be efficiently approximated by the Monte Carlo simulation for different architectures. The result of simulation can be used to calculate the reliability criterion for a wide range of injection rate and TSV failure rate values.

- **Novel dynamic 3D-specific TSV coupling fault model**

Chapter 4 proposed a system-level TSV-to-TSV coupling fault model that models the capacitive coupling effect, considering thermal impact, with circuit-level accuracy. This model can be plugged into any system-level TSV-based 3D-NoC simulator. It is also capable of identifying faulty TSV bundles and evaluating the efficiency of alternative resilient TSV-based 3D-NoC designs at the system-level. The presented model can be utilized for application-specific designs by addressing the susceptible to failure TSVs. With these results a designer is able to employ fault-tolerant methods only where they are required. For general purpose architectures, the presented fault model is able to figure out the effect of physical parameters of TSVs on timing requirement of the circuits. This model can be used to find the suitable physical parameters for a TSV to have reliable TSV links.

- **Fault-avoidance coding and architecture to avoid TSV coupling**

In Chapter 5, first the inductance parasitics in contemporary TSVs is characterized, and then a classification for inductive coupling cases is analyzed and presented. Next, a coding algorithm is devised to mitigate the TSV-to-TSV inductive coupling. The coding method controls the current flow direction in TSVs

by adjusting the data bit streams at runtime to minimize the inductive coupling effects. After performing formal analysis on the efficiency scalability of devised algorithm, an enhanced approach supporting larger bus sizes is proposed. The experimental results showed that the proposed coding algorithm yields significant improvements, while its hardware-implemented encoder results in tangible latency, power consumption, and area.

In Chapter 6, two coding methods, baseline and enhanced, is proposed in order to minimize the TSV-to-TSV capacitive coupling effect. Baseline algorithm is proposed for small mesh of TSVs which are considered in 3D-NoC applications, while enhanced method is suggested for large mesh of TSVs which are more applied in 3D memory applications. The enhanced method guarantees that the encoding process eliminates undesirable parasitic capacitance values by recognizing all susceptible configurations. According to experimental results, the baseline method's mitigation rate is more than 90% for TSV meshes smaller than  $10 \times 10$ . The enhanced algorithm mitigates the TSV-to-TSV capacitive coupling more than 70% for  $8 \times 32$  mesh of TSVs.

In Chapter 7, a novel architectural approach (hardware redundancy) is devised to effectively reduce the effect of TSV-to-TSV capacitive coupling. This approach, uses dual-rail coding combined with multiple lane QDI asynchronous SerDes to exploit the high-performance feature of TSV and compensate the inevitable extra bits imposed by dual-rail coding.

## **8.3 Future Work**

The potential future research work worth pursuing are categorized in the following areas:

### **8.3.1 Extending Dynamic TSV Coupling Fault Model**

The dynamic TSV coupling fault model can be expanded to support more than four aggressors which needs more complicated study and analyses, but it adds to the models accuracy.

### **8.3.2 Coupling Fault Avoidance or Recovery**

A study should be performed to compare the pros and cons of the proposed fault avoidance techniques with conventional fault-tolerant (recovery) approaches for TSV coupling fault. This study helps the DFT designers to choose the apposite method to make their 3D designs more reliable.

## **8.4 Concluding Remarks**

In conclusion, the work presented in this dissertation provides an insight to make the TSV-based 3D architectures more reliable. It helps the 3D-IC designers to improve the resiliency of their designs more effectively.



# Bibliography

- [1] TOP500 Supercomputer Site.
- [2] K. Aisopos, C.-H. Chen, and L.-S. Peh. Enabling system-level modeling of variation-induced faults in networks-on-chips. In *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, pages 930–935, 2011.
- [3] S. Akbari, A. Shafiee, M. Fathy, and R. Berangi. Afra: A low cost high performance reliable routing for 3d mesh nocs. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2012*, pages 332–337, March 2012.
- [4] H. Aliee, H. R. Zarandi, and P. M. Yaghini. K2router: A low-power and high-performance router design for networks-on-chip. *Journal on Computer Science and Engineering*, 7(2):8–23, 2011.
- [5] S. A. Asghari, H. Pedram, M. Khademi, and P. Yaghini. Designing and implementation of a network on chip router based on handshaking communication mechanism. *World Applied Sciences Journal*, 6(1):88–93, 2009.
- [6] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *Dependable and Secure Computing, IEEE Transactions on*, 1(1):11–33, Jan 2004.
- [7] L. T. B. Wu, X. Gu and M. Ritter. Electromagnetic modeling of massively coupled through silicon vias for 3d interconnects. In *Microwave and Optical Technology Letters*, page 12041206, 2011.
- [8] J. H. Bahn and N. Bagherzadeh. A generic traffic model for on-chip interconnection networks. *Network on Chip Architectures*, page 22, 2008.
- [9] R. C. BAUMANN. Soft errors in commercial integrated circuits. *International Journal of High Speed Electronics and Systems*, 14(02):299–309, 2004.
- [10] A. Benso and P. Prinetto. *Fault Injection Techniques and Tools for Embedded Systems Reliability Evaluation*. Springer Publishing Company, Incorporated, 1st edition, 2010.

- [11] C. Bienia and K. Li. Parsec 2.0: A new benchmark suite for chip-multiprocessors. In *Proceedings of the 5th Annual Workshop on Modeling, Benchmarking and Simulation*, June 2009.
- [12] C. Bienia and K. Li. Fidelity and scaling of the parsec benchmark inputs. In *Proceedings of the IEEE International Symposium on Workload Characterization (IISWC'10)*, IISWC '10, pages 1–10, Washington, DC, USA, 2010. IEEE Computer Society.
- [13] J. Borecky, M. Kohlik, P. Kubalik, and H. Kubatova. Fault models usability study for on-line tested fpga. In *Digital System Design (DSD), 2011 14th Euromicro Conference on*, pages 287–290, 2011.
- [14] S. Borkar. The exascale challenge. In *VLSI Design Automation and Test (VLSI-DAT), 2010 International Symposium on*, pages 2–3, 2010.
- [15] S. Borkar. 3d integration for energy efficient system design. In *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, pages 214–219, 2011.
- [16] D. Brooks, R. Dick, R. Joseph, and L. Shang. Power, thermal, and reliability modeling in nanometer-scale microprocessors. *Micro, IEEE*, 27(3):49–62, 2007.
- [17] C. Cassidy, J. Kraft, S. Carniello, F. Roger, H. Ceric, A. Singulani, E. Langer, and F. Schrank. Through silicon via reliability. *Device and Materials Reliability, IEEE Transactions on*, 12(2):285–295, 2012.
- [18] V. Champac, J. Hernandez, S. Barcelo, R. Gomez, C. Hawkins, and J. Segura. Testing of stuck-open faults in nanometer technologies. *Design Test of Computers, IEEE*, 29(4):80–91, 2012.
- [19] H.-M. Chang, J.-L. Huang, D.-M. Kwai, K.-T. Cheng, and C.-W. Wu. An error tolerance scheme for 3d cmos imagers. In *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, pages 917–922, June 2010.
- [20] Y.-C. Chang, C.-T. Chiu, S.-Y. Lin, and C.-K. Liu. On the design and analysis of fault tolerant noc architecture using spare routers. In *Design Automation Conference (ASP-DAC), 2011 16th Asia and South Pacific*, pages 431–436, 2011.
- [21] J. Chen, D.-W. Chang, C.-P. Young, G.-Y. Huang, S.-L. Chu, C.-Y. Ke, S.-T. Yen, and T.-S. Kuo. Building a multi-kernel embedded system with high performance ipc mechanism. In *High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on*, pages 506–511, sept. 2011.

- [22] Y. Cheng, L. Zhang, Y. Han, and X. Li. Thermal-constrained task allocation for interconnect energy reduction in 3-d homogeneous mpsocs. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 21(2):239–249, 2013.
- [23] A. Dalirsani, M. Hosseinabady, and Z. Navabi. An analytical model for reliability evaluation of noc architectures. In *On-Line Testing Symposium, 2007. IOLTS 07. 13th IEEE International*, pages 49–56, July.
- [24] F. Darve, A. Sheibanyrad, P. Vivet, and F. Petrot. Physical implementation of an asynchronous 3d-noc router using serial vertical links. In *VLSI (ISVLSI), 2011 IEEE Computer Society Annual Symposium on*, pages 25–30, July 2011.
- [25] A. DeOrio, D. Fick, V. Bertacco, D. Sylvester, D. Blaauw, J. Hu, and G. Chen. A reliable routing architecture and algorithm for nocs. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 31(5):726–739, May.
- [26] F. Dubois, A. Sheibanyrad, F. Petrot, and M. Bahmani. Elevator-first: A deadlock-free distributed routing algorithm for vertically partially connected 3d-nocs. *Computers, IEEE Transactions on*, 62(3):609–615, 2013.
- [27] A. Eghbal, P. M. Yaghini, N. Bagherzadeh, and M. Khayambashi. Tsv analytical fault tolerance assessment for 3d network-on-chip. *Computers, IEEE Transactions on*, 2015.
- [28] A. Eghbal, P. Yaghini, S. Yazdi, and N. Bagherzadeh. Tsv-to-tsv inductive coupling-aware coding scheme for 3d network-on-chip. In *Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2014 IEEE International Symposium on*, pages 92–97, Oct 2014.
- [29] A. Eghbal, P. M. Yaghini, and N. Bagherzadeh. Capacitive coupling mitigation for tsv-based 3d ics. In *2015 IEEE 33rd VLSI Test Symposium (VTS)*, pages 1–6, April 2015.
- [30] A. Eghbal, P. M. Yaghini, N. Bagherzadeh, and M. Khayambashi. Analytical fault tolerance assessment and metrics for tsv-based 3d network-on-chip. 2015.
- [31] A. Eghbal, P. M. Yaghini, H. Pedram, and H. R. Zarandi. Fault injection-based evaluation of a synchronous noc router. In *On-Line Testing Symposium, 2009. IOLTS 2009. 15th IEEE International*, pages 212–214. IEEE, 2009.
- [32] A. Eghbal, P. M. Yaghini, H. Pedram, and H. R. Zarandi. Designing fault-tolerant network-on-chip router architecture. *International Journal of Electronics*, 97(10):1181–1192, 2010.

- [33] A. Eghbal, H. R. Zarandi, and P. M. Yaghini. Fault tolerance assessment of pic microcontroller based on fault injection. In *Test Workshop, 2009. LATW'09. 10th Latin American*, pages 1–6. IEEE, 2009.
- [34] G. S. for Microelectronics Industry. Failure mechanisms and models for semiconductor devices. *JEDEC Publication JEP 122-G*, page 0, 2011.
- [35] R. Gaillard. Single event effects: Mechanisms and classification. In M. Nicolaidis, editor, *Soft Errors in Modern Electronic Systems*, volume 41 of *Frontiers in Electronic Testing*, pages 27–54. Springer US, 2011.
- [36] A. Ghiribaldi, A. Strano, M. Favalli, and D. Bertozzi. Power efficiency of switch architecture extensions for fault tolerant noc design. In *Green Computing Conference (IGCC), 2012 International*, pages 1–6, 2012.
- [37] D. Gil-Tomas, J. Gracia-Moran, J.-C. Baraza-Calvo, L.-J. Saiz-Adalid, and P.-J. Gil-Vicente. Analyzing the impact of intermittent faults on microprocessors applying fault injection. *Design Test of Computers, IEEE*, 29(6):66–73, 2012.
- [38] <http://www.itrs.net/Links/2011ITRS/Home2011.htm>. *ITRS 2011*, 2011.
- [39] Intel-cooperation. Pin - A Dynamic Binary Instrumentation Tool.
- [40] K. Ishida, T. Yasufuku, S. Miyamoto, H. Nakai, M. Takamiya, T. Sakurai, and K. Takeuchi. A 1.8v 30nj adaptive program-voltage (20v) generator for 3d-integrated nand flash ssd. In *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pages 238 –239,239a, feb. 2009.
- [41] K. Ishida, T. Yasufuku, S. Miyamoto, H. Nakai, M. Takamiya, T. Sakurai, and K. Takeuchi. A 1.8v 30nj adaptive program-voltage (20v) generator for 3d-integrated nand flash ssd. In *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pages 238–239,239a, 2009.
- [42] S. Itr. ITRS 2012 Executive Summary.
- [43] S. Itr. Table intc7 - itrs 2013 executive summary.
- [44] H. Iwai. Roadmap for 22nm and beyond (invited paper). *Microelectron. Eng.*, 86(7-9):1520–1528, July 2009.
- [45] J. Jeddloh and B. Keeth. Hybrid memory cube new dram architecture increases density and performance. In *VLSI Technology (VLSIT), 2012 Symposium on*, pages 87–88, June 2012.

- [46] N. Julai, A. Yakovlev, and A. Bystrov. Error detection and correction of single event upset (seu) tolerant latch. In *On-Line Testing Symposium (IOLTS), 2012 IEEE 18th International*, pages 1–6, 2012.
- [47] M. Jung, J. Mitra, D. Pan, and S. K. Lim. Tsv stress-aware full-chip mechanical reliability analysis and optimization for 3-d ic. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 31(8):1194–1207, 2012.
- [48] A. Kamto, Y. Liu, L. Schaper, and S. Burkett. Reliability study of through-silicon via (tsv) copper filled interconnects. *Thin Solid Films*, 518(5):1614 – 1619, 2009. Proceedings of the 36th International Conference on Metallurgical Coatings and Thin Films.
- [49] U. Kang, H.-J. Chung, S. Heo, D.-H. Park, H. Lee, J. H. Kim, S.-H. Ahn, S.-H. Cha, J. Ahn, D. Kwon, J.-W. Lee, H.-S. Joo, W.-S. Kim, D. H. Jang, N. S. Kim, J.-H. Choi, T.-G. Chung, J.-H. Yoo, J. S. Choi, C. Kim, and Y.-H. Jun. 8 gb 3-d ddr3 dram using through-silicon-via technology. *Solid-State Circuits, IEEE Journal of*, 45(1):111 –119, jan. 2010.
- [50] U. Kang, H.-J. Chung, S. Heo, D.-H. Park, H. Lee, J.-H. Kim, S.-H. Ahn, S.-H. Cha, J. Ahn, D. Kwon, J.-W. Lee, H.-S. Joo, W.-S. Kim, D. H. Jang, N. S. Kim, J.-H. Choi, T.-G. Chung, J.-H. Yoo, J.-S. Choi, C. Kim, and Y.-H. Jun. 8 gb 3-d ddr3 dram using through-silicon-via technology. *Solid-State Circuits, IEEE Journal of*, 45(1):111–119, 2010.
- [51] G. Katti, M. Stucchi, K. De Meyer, and W. Dehaene. Electrical modeling and characterization of through silicon via for three-dimensional ics. *Electron Devices, IEEE Transactions on*, 57(1):256 –262, jan. 2010.
- [52] A. Kauppila, G. Vaughn, J. Kauppila, and L. Massengill. Probabilistic evaluation of analog single event transients. *Nuclear Science, IEEE Transactions on*, 54(6):2131–2136, 2007.
- [53] M. Khayambashi, P. M. Yaghini, A. Eghbal, and N. Bagherzadeh. Analytical reliability analysis of 3d noc under tsv failure. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 11(4):43, 2015.
- [54] D. H. Kim and S.-K. Lim. Design quality trade-off studies for 3-d ics built with sub-micron tsvs and future devices. *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, 2(2):240–248, 2012.
- [55] D. H. Kim, S. Mukhopadhyay, and S.-K. Lim. Fast and accurate analytical modeling of through-silicon-via capacitive coupling. *Components, Packaging and Manufacturing Technology, IEEE Transactions on*, 1(2):168–180, Feb 2011.

- [56] M. Koibuchi, H. Matsutani, H. Amano, and T. Pinkston. A lightweight fault-tolerant mechanism for network-on-chip. In *Networks-on-Chip, 2008. NoCS 2008. Second ACM/IEEE International Symposium on*, pages 13–22, April.
- [57] M. Krasich. How to estimate and use mttf/mtbf would the real mtbf please stand up? In *Reliability and Maintainability Symposium. Annual*, pages 353–359, Jan.
- [58] R. Kumar and S. P. Khatri. Crosstalk avoidance codes for 3d vlsi. In *Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1673–1678, March 2013.
- [59] S. Kumar and R. Van Leuken. A 3d network-on-chip for stacked-die transactional chip multiprocessors using through silicon vias. In *Design Technology of Integrated Systems in Nanoscale Era (DTIS), 2011 6th International Conference on*, pages 1–6, April 2011.
- [60] J. Lee, D. Lee, S. Kim, and K. Choi. Deflection routing in 3d network-on-chip with tsv serialization. In *Design Automation Conference (ASP-DAC), 2013 18th Asia and South Pacific*, pages 29–34, Jan 2013.
- [61] M. Lee. *Electrical Design of Through Silicon Via*. SpringerLink : Bücher. Springer London, Limited, 2014.
- [62] C. Liu, T. Song, J. Cho, J. Kim, J. Kim, and S.-K. Lim. Full-chip tsv-to-tsv coupling analysis and optimization in 3d ic. In *Design Automation Conference, 48th ACM/EDAC/IEEE*, pages 783–788, 2011.
- [63] X. Liu, Q. Chen, P. Dixit, R. Chatterjee, R. Tummala, and S. Sitaraman. Failure mechanisms and optimum design for electroplated copper through-silicon vias (tsv). In *Electronic Components and Technology Conference, 2009. ECTC 2009. 59th*, pages 624–629, 2009.
- [64] I. Loi, F. Angiolini, S. Fujita, S. Mitra, and L. Benini. Characterization and implementation of fault-tolerant vertical links for 3-d networks-on-chip. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 30(1):124–134, 2011.
- [65] F. Lu, J. Shao, X. Liu, and X. Wang. Validation test method of tddb physics-of-failure models. In *Prognostics and System Health Management (PHM), 2012 IEEE Conference on*, pages 1–4, 2012.
- [66] K. Lu, S.-K. Ryu, Q. Zhao, X. Zhang, J. Im, R. Huang, and P. S. Ho. Thermal stress induced delamination of through silicon vias in 3-d interconnects. In *Electronic Components and Technology Conference (ECTC), Proceedings 60th*, pages 40–45, 2010.

- [67] K. Lu, X. Zhang, S.-K. Ryu, J. Im, R. Huang, and P. S. Ho. Thermo-mechanical reliability of 3-d ics containing through silicon vias. In *Electronic Components and Technology Conference, 2009. ECTC 2009. 59th*, pages 630–634, 2009.
- [68] Z. Lu and A. Jantsch. Trends of terascale computing chips in the next ten years. In *ASIC, 2009. ASICON '09. IEEE 8th International Conference on*, pages 62–66, oct. 2009.
- [69] M. Lundstrom and J. Guo. *Nanoscale Transistors: Device Physics, Modeling and Simulation*. Springer, 2006.
- [70] R. Marculescu, U. Ogras, L.-S. Peh, N. Jerger, and Y. Hoskote. Outstanding research problems in noc design: System, microarchitecture, and circuit perspectives. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 28(1):3–21, 2009.
- [71] S. Marella, S. Kumar, and S. Sapatnekar. A holistic analysis of circuit timing variations in 3d-ics with thermal and tsv-induced stress considerations. In *Computer-Aided Design (ICCAD), 2012 IEEE/ACM International Conference on*, pages 317–324, 2012.
- [72] A. Mercha et al. Comprehensive analysis of the impact of single and arrays of through silicon vias induced stress on high-k / metal gate cmos performance. In *Electron Devices Meeting (IEDM), 2010 IEEE International*, pages 2.2.1–2.2.4, 2010.
- [73] Micron. Hybrid memory cube.
- [74] A. Mokhov, V. Khomenko, D. Sokolov, and A. Yakovlev. On dual-rail control logic for enhanced circuit robustness. In *Application of Concurrency to System Design (ACSD), 2012 12th International Conference on*, pages 112–121, June 2012.
- [75] R. Moonen, P. Vanmeerbeek, G. Lekens, W. De Ceuninck, P. Moens, and J. Boutsen. Study of time-dependent dielectric breakdown on gate oxide capacitors at high temperature. In *Physical and Failure Analysis of Integrated Circuits, 2007. IPFA 2007. 14th International Symposium on the*, pages 288–291, 2007.
- [76] A. Munir, S. Ranka, and A. Gordon-Ross. High-performance energy-efficient multicore embedded computing. *Parallel and Distributed Systems, IEEE Transactions on*, 23(4):684–700, april 2012.
- [77] A. Munir, S. Ranka, and A. Gordon-Ross. High-performance energy-efficient multicore embedded computing. *Parallel and Distributed Systems, IEEE Transactions on*, 23(4):684–700, april 2012.

- [78] A. Munir, S. Ranka, and A. Gordon-Ross. High-performance energy-efficient multicore embedded computing. *Parallel and Distributed Systems, IEEE Transactions on*, 23(4):684–700, 2012.
- [79] T. A. S. on Exascale Computing. *The Opportunities and Challenges of Exascale Computing*, 2010.
- [80] S. Park, T. Krishna, C.-H. Chen, B. Daya, A. Chandrakasan, and L.-S. Peh. Approaching the theoretical limits of a mesh noc with a 16-node chip prototype in 45nm soi. In *Proceedings of the 49th Annual Design Automation Conference, DAC '12*, pages 398–405, New York, NY, USA, 2012. ACM.
- [81] V. Pasca, L. Anghel, C. Rusu, and M. Benabdenbi. Configurable serial fault-tolerant link for communication in 3d integrated systems. In *On-Line Testing Symposium (IOLTS), 2010 IEEE 16th International*, pages 115–120, 2010.
- [82] S. Pasricha and Y. Zou. A low overhead fault tolerant routing scheme for 3d networks-on-chip. In *Quality Electronic Design (ISQED), 2011 12th International Symposium on*, pages 1–8, March.
- [83] Y. Peng, T. Song, D. Petranovic, and S. K. Lim. On accurate full-chip extraction and optimization of tsv-to-tsv coupling elements in 3d ics. In *Computer-Aided Design (ICCAD), 2013 IEEE/ACM International Conference on*, pages 281–288, Nov 2013.
- [84] Y. Peng, T. Song, D. Petranovic, and S. K. Lim. Silicon effect-aware full-chip extraction and mitigation of tsv-to-tsv coupling. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 33(12):1900–1913, Dec 2014.
- [85] I. Pomeranz and S. Reddy. A bridging fault model where undetectable faults imply logic redundancy. In *Design, Automation and Test in Europe, 2008. DATE '08*, pages 1166–1171, 2008.
- [86] I. Pomeranz and S. Reddy. Transition path delay faults: A new path delay fault model for small and large delay defects. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 16(1):98–107, 2008.
- [87] J. Pontes, N. Calazans, and P. Vivet. An accurate single event effect digital design flow for reliable system level design. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2012*, pages 224–229, 2012.
- [88] PTM. Predictive Technology Model.



- [89] A. M. Rahmani, K. R. Vaddina, K. Latif, P. Liljeberg, J. Plosila, and H. Tenhunen. Design and management of high-performance, reliable and thermal-aware 3d networks-on-chip. *Circuits, Devices Systems, IET*, 6(5):308–321, 2012.
- [90] N. Ranganathan, K. Prasad, N. Balasubramanian, and K. L. Pey. A study of thermo-mechanical stress and its impact on through-silicon vias. *Journal of Micromechanics and Microengineering*, 18(7):075018, 2008.
- [91] P. Roth, C. Jacobi, and K. Weber. Superprocessors and supercomputers. In B. Hoefflinger, editor, *Chips 2020*, The Frontiers Collection, pages 421–427. Springer Berlin Heidelberg, 2012.
- [92] S. Roy, C. Giri, S. Ghosh, and H. Rahaman. Optimizing test wrapper for embedded cores using tsv based 3d socs. In *VLSI (ISVLSI), 2011 IEEE Computer Society Annual Symposium on*, pages 31–36, July 2011.
- [93] S.-K. Ryu, K.-H. Lu, T. Jiang, J.-H. Im, R. Huang, and P. S. Ho. Effect of thermal stresses on carrier mobility and keep-out zone around through-silicon vias for 3-d integration. *Device and Materials Reliability, IEEE Transactions on*, 12(2):255–262, 2012.
- [94] I. Savidis and E. Friedman. Closed-form expressions of 3-d via resistance, inductance, and capacitance. *Electron Devices, IEEE Transactions on*, 56(9):1873–1881, 2009.
- [95] C. Selvanayagam, J. Lau, X. Zhang, S. K. W. Seah, K. Vaidyanathan, and T. Chai. Nonlinear thermal stress/strain analyses of copper filled tsv (through silicon via) and their flip-chip microbumps. In *Electronic Components and Technology Conference, 2008. ECTC 2008. 58th*, pages 1073–1081, 2008.
- [96] A. Shayan, X. Hu, H. Peng, C.-K. Cheng, W. Yu, M. Popovich, T. Toms, and X. Chen. Reliability aware through silicon via planning for 3d stacked ics. In *Design, Automation Test in Europe Conference Exhibition, 2009. DATE '09.*, pages 288–291, 2009.
- [97] A. Shayan, X. Hu, W. Zhang, C.-K. Cheng, A. Ege Engin, X. Chen, and M. Popovich. 3d stacked power distribution considering substrate coupling. In *Computer Design. IEEE International Conference on*, pages 225–230, 2009.
- [98] T. Song, C. Liu, Y. Peng, and S. K. Lim. Full-chip multiple tsv-to-tsv coupling extraction and optimization in 3d ics. In *Design Automation Conference (DAC), 2013 50th ACM / EDAC / IEEE*, pages 1–7, May 2013.

- [99] M. Stanisavljevi, A. Schmid, and Y. Leblebici. Reliability, faults, and fault tolerance. In *Reliability of Nanoscale Circuits and Systems*, pages 7–18. Springer New York, 2011.
- [100] V. Suntharalingam, R. Berger, S. Clark, J. Knecht, A. Messier, K. Newcomb, D. Rathman, R. Slattery, A. Soares, C. Stevenson, K. Warner, D. Young, L. P. Ang, B. Mansoorian, and D. Shaver. A 4-side tileable back illuminated 3d-integrated mpixel cmos image sensor. In *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pages 38 –39,39a, feb. 2009.
- [101] J. T. Pawlowski. Hybrid memory cube (hmc). In *HOT-CHIPS*, 2011.
- [102] THENoC. 3d noc simulator.
- [103] A. Todri, S. Kundu, P. Girard, A. Bosio, L. Dilillo, and A. Virazel. A study of tapered 3-d tsvs for power and thermal integrity. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 21(2):306–319, 2013.
- [104] W.-C. Tsai, D.-Y. Zheng, S.-J. Chen, and Y.-H. Hu. A fault-tolerant noc scheme using bidirectional channel. In *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, pages 918–923, June.
- [105] K. Tu. Reliability challenges in 3d ic packaging technology. *Microelectronics Reliability*, 51(3):517 – 523, 2011.
- [106] R. Ubar, S. Kostin, and J. Raik. Multiple stuck-at-fault detection theorem. In *Design and Diagnostics of Electronic Circuits Systems (DDECS), 2012 IEEE 15th International Symposium on*, pages 236–241, 2012.
- [107] G. Van der Plas, P. Limaye, A. Mercha, H. Oprins, C. Torregiani, S. Thijs, D. Linten, M. Stucchi, K. Guruprasad, D. Velenis, D. Shinichi, V. Cherman, B. Vandeveld, V. Simons, I. De Wolf, R. Labie, D. Perry, S. Bronckers, N. Minas, M. Cupac, W. Ruythooren, J. Van Olmen, A. Phommahaxay, M. de Potter de ten Broeck, A. Opdebeeck, M. Rakowski, B. De Wachter, M. Dehan, M. Nelis, R. Agarwal, W. Dehaene, Y. Travaly, P. Marchal, and E. Beyne. Design issues and considerations for low-cost 3d tsv ic technology. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pages 148 –149, feb. 2010.
- [108] J. e. Warnock. A 5.2ghz microprocessor chip for the ibm zenterprise system. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pages 70–72, 2011.

- [109] R. Weerasekera, M. Grange, D. Pamunuwa, H. Tenhunen, and L.-R. Zheng. Compact modelling of through-silicon vias (tsvs) in three-dimensional (3-d) integrated circuits. In *3D System Integration, 2009. 3DIC 2009. IEEE International Conference on*, pages 1–8, 2009.
- [110] B. Wu, X. Gu, L. Tsang, and M. B. Ritter. Electromagnetic modeling of massively coupled through silicon vias for 3d interconnects. *Microwave and Optical Technology Letters*, 53(6):1204–1206, 2011.
- [111] T. Xu, P. Liljeberg, and H. Tenhunen. A study of through silicon via impact to 3d network-on-chip design. In *Electronics and Information Engineering (ICEIE), 2010 International Conference On*, volume 1, pages V1–333–V1–337, Aug 2010.
- [112] P. Yaghini, A. Eghbal, M. Khayambashi, and N. Bagherzadeh. Coupling mitigation in 3-d multiple-stacked devices. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, PP(99):1–1, 2015.
- [113] P. M. Yaghini, A. Eghbal, S. Asghari, and H. Pedram. Power comparison of an asynchronous and synchronous network on chip router. *Computer Conference*, pages 242–246, 2009.
- [114] P. M. Yaghini, A. Eghbal, and N. Bagherzadeh. A gals router for asynchronous network-on-chip. In *Proceedings of International Workshop on Manycore Embedded Systems, MES '14*, pages 52:52–52:55, New York, NY, USA, 2014. ACM.
- [115] P. M. Yaghini, A. Eghbal, and N. Bagherzadeh. On the design of hybrid routing mechanism for mesh-based network-on-chip. *Integration, the {VLSI} Journal*, 50:183 – 192, 2015.
- [116] P. M. Yaghini, A. Eghbal, H. Pedram, and H. R. Zarandi. Investigation of transient fault effects in an asynchronous noc router. In *Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on*, pages 540–545. IEEE, 2010.
- [117] P. M. Yaghini, A. Eghbal, H. Pedram, and H. R. Zarandi. Investigation of transient fault effects in synchronous and asynchronous network on chip router. *J. Syst. Archit.*, 57(1):61–68, Jan. 2011.
- [118] P. M. Yaghini, A. Eghbal, S. S. Yazdi, and N. Bagherzadeh. Accurate system-level tsv-to-tsv capacitive coupling fault model for 3d-noc. In *Proceedings of the 9th International Symposium on Networks-on-Chip*, page 3. ACM, 2015.
- [119] P. M. Yaghini, A. Eghbal, S. S. Yazdi, N. Bagherzadeh, and M. M. Green. Capacitive and inductive tsv-to-tsv resilient approaches for 3d ics. *IEEE Transactions on Computers*, 65(3):693–705, March 2016.

- [120] P. M. Yaghini, H. R. Zarandi, A. Eghbal, A. Jafarzadeh, and S. Eskandari. An investigation of fault tolerance behavior of 32-bit dlx processor. In *Dependability, 2009. DEPEND'09. Second International Conference on*, pages 93–98. IEEE, 2009.
- [121] X. Yang, Z. Wang, J. Xue, and Y. Zhou. The reliability wall for exascale supercomputing. *Computers, IEEE Transactions on*, 61(6):767–779, 2012.
- [122] H. Yoshikawa, A. Kawasaki, Tomoaki, Iiduka, Y. Nishimura, K. Tanida, K. Akiyama, M. Sekiguchi, M. Matsuo, S. Fukuchi, and K. Takahashi. Chip scale camera module (cscm) using through-silicon-via (tsv). In *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pages 476–477,477a, feb. 2009.
- [123] W. Zhang and T. Li. Microarchitecture soft error vulnerability characterization and mitigation under 3d integration technology. In *Microarchitecture. MICRO-41. 2008 41st IEEE/ACM International Symposium on*, pages 435–446, 2008.
- [124] M. Zhu, J. Lee, and K. Choi. An adaptive routing algorithm for 3d mesh noc with limited vertical bandwidth. In *VLSI and System-on-Chip (VLSI-SoC), 2012 IEEE/IFIP 20th International Conference on*, pages 18–23, Oct 2012.