UNIVERSITY OF CALIFORNIA,
IRVINE

Tackling the Effects of Elevated Temperature and Aging Phenomena in 3D Integrated Circuits

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical and Computer Engineering

by

Ayed Saad A Alqahtani

Dissertation Committee:
Professor Nader Bagherzadeh, Chair
Professor Rainer Dömer
Professor Alexander Veidenbaum

2019

DEDICATION

This dissertation is dedicated to:
my family
for their love and support

وَمَنْ يَتَهَيَّب صُعُودَ الجِبَـالِ
يَعِشْ أَبَدَ الدَهرِ بَيْنَ الحُفَـر

"Those who do not like climbing the mountains will live forever among the hollows", from "The will of life" poem, by Abi Alqasim Alshabbi (1909-1934).

A person who is afraid of trying, will not be able to achieve any success throughout his life. If you are afraid to climb mountains, you will not be able to get out of your holes. One who wants to survive and get up from these holes should not be afraid. One who wants to advance must learn that climbing mountains is the way to success.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

I am extremely grateful to God Almighty who bestowed upon me the understanding, perseverance and critical thinking to make this accomplishment possible.

I would like to express my appreciation and thanks to my dissertation chairman and adviser, Professor Nader Bagherzadeh, for his incessant guidance through out dissertation completion process. He has been available whenever I needed his assistance. I really appreciate his help in developing ideas as well as his positive comments for improving and bringing out the utmost optimum consequences out of my endeavors. Without his guidance and support, I would have never been able to organize and complete the dissertation tasks as optimal and within time constraints.

I would like to extend my deep appreciation to *King Saud University* (KSU) and the *Saudi Arabian Cultural Mission* (SACM) for their support, funds, and assistance. Also, I would like to express my thanks and appreciation to the dissertation committee members Professor Rainer Dömer and Professor Alexander Veidenbaum for their creative comments and guidance through out the dissertation completion.

In addition, I express my greatest regards to my friends and colleagues for their support throughout my stay at UCI. They have been a steady source for help and encouragement.

Last but not least, I am highly indebted to all my family members for their love and patience. Without their support I would have never been able to finish the dissertation.

Ayed

April 2019

CURRICULUM VITAE

Ayed Saad A Alqahtani

EDUCATION

Doctor of Philosophy in Electrical and Computer Engineering **2019**
Univercity of California, Irvine (UCI) *Irvine, California*

Master of Science in Computer & Information Sciences **2010**
King Fahad University for Petroleum & Minerals (KFUPM) *Dhahran, Saudi Arabia*

Bachelor of Science in Computer & Information Sciences **2007**
King Saud University (KSU) *Riyadh, Saudi Arabia*

WORK EXPERIENCE

Consultant **Sep 2006–Jan 2008**
Al-ELM Information Security Company *Riyadh, Saudi Arabia*

R&D Engineer **Jun 2007–Oct 2007**
Advanced Electronic Company (AEC) *Riyadh, Saudi Arabia*

Smart Card Developer **Jun 2006–Sep 2006**
Al-ELM Information Security Company *Riyadh, Saudi Arabia*

Testing Engineer **Summer 2005**
Advanced Electronic Company (AEC) *Riyadh, Saudi Arabia*

TEACHING EXPERIENCE

Lecturer **2010–2014**
VLSI Design and Testing,

Control Systems,
Computer Networks,
Computer Security
King Saud University (KSU) *Riyadh, Saudi Arabia*
Teaching Assistant **2006–2010**
VLSI Design and Testing,

Control Systems
King Saud University (KSU) *Riyadh, Saudi Arabia*

REFEREED JOURNAL PUBLICATIONS

System Level Analysis of 3D ICs with Thermal TSVs **2018**
ACM Journal on Emerging Technologies in Computing (JETC)

AROMa: Aging-Aware Deadlock-Free Adaptive Routing Algorithm and Online Monitoring in 3D NoCs **2017**
IEEE Transactions on Parallel & Distributed Systems (TPDS)

A Finite State Machine Based Fault Tolerance Technique for Sequential Circuits **2013**
Microelectronics Reliability

REFEREED CONFERENCE PUBLICATIONS

Thermal Analysis of 3D ICs with TSVs Using System Level Simulations **2018**
Semiconductor Research Corporation (SRC) TECHCON 2018

Online monitoring and adaptive routing for aging mitigation in NoCs **2017**
Design, Automation & Test in Europe Conference & Exhibition (DATE)

Serial vs. parallel elliptic curve crypto processor designs **2013**
IADIS International Conference: Applied Computing

Triple-A: Secure RGB Image Steganography Based on Randomization **2009**
The 7th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)

Asymptotic Behavior of Networked Control System (NCS) **2007**
18th National Computer Conference (NCC18)

ABSTRACT OF THE DISSERTATION

Tackling the Effects of Elevated Temperature and Aging Phenomena in 3D Integrated Circuits

By

Ayed Saad A Alqahtani

Doctor of Philosophy in Electrical and Computer Engineering

University of California, Irvine, 2019

Professor Nader Bagherzadeh, Chair

With technology advancement to the nanoscale level, 3D stacking of *Integrated Circuits* (ICs) provides significant advantages in saving device footprints, improving power management, and continuing performance enhancement which aim to increase energy efficiency and scalability particularly for many-core and NoCs systems. However, the benefits of these systems can be jeopardized because they became more subjected to elevated temperature induced by thermal management challenges as well as delay degradation (i.e. aging) caused by load imbalance.

On one hand, the elevated temperature in 3D systems can be solved by *Thermal Through Silicon Vias* (TTSVs). However, past research has either overestimated or underestimated the effects of TTSVs as a consequence of the lack of detailed 3D IC models or system-level simulations. In this dissertation, we propose a simulation flow to accurately simulate TTSV effects on 3D ICs. Furthermore, we present a hierarchical approach to optimize the floorplan of a 3D Nehalem-based multicore processor via *Simulated Annealing* (SA) with respect to the area, temperature, and wirelength. By using detailed 3D thermal model along with full system and validated thermal simulators, our results show accurate thermal analysis of 3D ICs. In addition, we found that the peak temperature of 3D IC such as 3D Nehalem is reduced with a minimal area overhead.

On the other hand, previous mitigation techniques to reduce aging phenomena effects on NoC system either ignore the runtime operating conditions or impose significant overheads to the system. Hence, this dissertation also presents an online monitoring method through a *Centralized Aging Table* (CAT) for routers in NoCs. Our methodology populates CAT by values that represent aging degradation for each different pairs of stress and temperature ranges during a given period of time. Moreover, utilizing CAT, we propose an online adaptive aging-aware routing algorithm in order to avoid highly aged routers in 2D NoC which eventually leads to age and load balancing between routers. We also extend this idea to 3D NoC by proposing AROMa, which is an aging-aware deadlock-free adaptive routing algorithm integrated with a novel online aging monitoring system for 3D NoCs. The monitoring system in AROMa exploits *Distributed-Centralized-Aging-Table* (D-CAT) to obtain routers' aging rates for each layer of 3D NoCs. Consequently, AROMa swaps between different k-best source-destination shortest paths periodically to avoid highly aged routers, force them in the recovery phase, and accordingly balance aging in the network. Our results demonstrate that our online routing algorithm and monitoring methodology improves delay degradation of maximum aged router and aging imbalance while ensuring that the impact of our proposed methodology on network latency, *Energy-Delay-Product* (EDP) and link utilization is negligible.

Keywords: *Multicore, Manycore, 3D ICs, Thermal Issues, Thermal Management, Through Silicon Via (TSV), Thermal Through Silicon Via (TTSV), Floorplanning, Network on Chips (NoCs), 3D NoCs, Aging, Delay Degradation; Online Monitoring; Adaptive Routing, Deadlock Freedom.*

Chapter 1

Introduction

In this chapter, we describe the importance of overcoming the effects of elevated temperature and aging phenomena in 3D Integrated Circuits (ICs), and briefly characterize delay degradation and Thermal Through Silicon Vias (TTSVs) in 3D stacked structures. Methodologies and techniques to reduce the issues associated with stacking integrated circuits especially temperature and delay degradation(i.e. aging) is emphasized as the topic of research for this dissertation. We specify the problem to be solved and list the dissertation objectives in the following subsections, the details of which is going to be demonstrated in their respective chapters. Before discussing several aspects related to the dissertation topic, we give an overview of the previous work in this field. In the following chapters, the background material are reinstated whenever relevant to the subject. Finally, we summarize the organization of this dissertation.

1.1 Elevated Temperature and Aging Phenomena in 3D ICs and NoCs

With technology advancement to the nanoscale level, 3D stacking of integrated circuits (ICs) provides significant advantages in saving device footprints, improving power management, and continuing performance enhancement which aim to increase energy efficiency and scalability particularly for many-core and NoCs systems [24, 85, 81, 50]. However, the benefits of these systems can be jeopardized because they became more subjected to elevated temperature induced by thermal management challenges as well as delay degradation (i.e. aging) caused by load imbalance [129].

1.1.1 Elevated Temperature Challenge in 3D ICs

The continual cramming of more silicon transistors onto chips, known as Moore's Law, has been the building block of exuberant innovation in computing. Now it is approaching its physical limitations, where transistors' density increase on a microprocessor is becoming intractable. To extend Moore's law life expectancy, 3D ICs offer one of the most practical solutions by expanding the architecture in the third dimension to reduce chip area and shorten the metal wires [24, 85, 81]. Die-to-Die (D2D) connections along with Through Silicon Vias (TSVs) made stacking of chips feasible. The D2D layers are made of microbumps (or μbumps) and underfill, which exist between dies for physical connections. TSVs run across the bulk silicon die, which serve as electrical connections between active layers [27]. Hence, the move from 2D to 3D stacking for System-on-Chip (SoC) and Network-on-Chip (NoC) introduces a methodology for integrating a very high number of logic in a single die. The achievable performance benefits arising out of adopting 3D stacking in these systems include performance gain, functionality, packaging density, and heterogeneous integration, especially in terms of throughput, latency, energy dissipation, and wiring area overhead reduction [52, 33, 110].

3D ICs attribute clear performance improvement as compared to 2D ICs, however, the thermal issues have become a severe problem as a direct result of the high power density and stacked structures. The power consumption of a 3D IC is expected to decrease because of the shorter interconnects, but, the power density increases induced by the high number of active devices per unit volume [33]. The stacked dies in 3D ICs are also detrimental to thermal management. For 2D ICs, the heat generated by the active circuits can be easily accessed and dissipated through the bulk silicon substrate and *Thermal Interface Material* (TIM) to the heat sink structures, which undergoes convection heat transfer to 25°C ambient air. Only a small fraction of the heat flows downwards through the *Back-End-Of-Line* (BEOL) wiring layers and the solider balls into the package substrate because of this path's high thermal resistance. On the other hand, for 3D ICs, there is no direct access to the stacked chips nor a method of dissipating heat laterally from the stacked dies, both the BEOL wiring layers and the μbump layer between the stacked chips are now in the thermal path [36]. The increased power density and the various low thermal conductivity interlayer materials have made it significantly more difficult to maintain all the dies at an acceptable junction temperature. The challenges in thermal management can be even higher because of the non-uniform power dissipation, which leads to the formation of local hot spots that cannot be easily eliminated by conventional heat sink structures [15, 112]. 3D ICs encounter inevitable thermal management problems [72, 128, 109, 16]. The major reasons for the thermal issues include high power density generated by a large number of active layers per unit area as well as high thermal resistance between active dies and heat sinks in a stacked structure. Moreover, the aggressive wafer thinning process makes the thermal management of 3D ICs more challenging [112].

1.1.2 Delay Degradation Problem in NoCs

The elevated temperature in 3D structures has a direct impact in delay degradation, i.e. aging, on the 3D chip. Delay degradation caused by aging mechanisms becomes a reliability challenge in advanced semiconductor technology [2]. It imposes a large design margin to the critical paths which results in design complexity and overhead [92, 75]. *Bias Temperature Instability* (BTI) and *Hot Carrier Injection* (HCI) are the dominant aging mechanisms that gradually increase the threshold voltage (V_{th}) of transistors [2, 43, 55, 58]. The shifted V_{th} leads to undesired increase in system critical path delay which ultimately exacerbates performance loss and timing failure within the system components in the long run. This impacts the lifetime of the chip in the long term and its performance in the short term. Consequently, designers have to allocate considerable guardband to the critical path for avoiding timing failure. This imposes power, area, and performance overheads to the system [35, 127, 97] and threaten the performance and scalability for many-core designs and NoCs which motivates careful aging investigation. Furthermore, critical path's age is affected by operating conditions such as stress (i.e. usage of transistors along it) and temperature, which change with time because of variation in running all applications on a system. In other words, higher temperature and stress leads to higher aging rate. Since flits are the only router's stimuli inside the router and their residence time affect the temperature and stress of the router, by monitoring them one can predict temperature and stress, which impact the aging rate. Additionally, the number of flits and their associated resident time are stimulated by the routing algorithm. Router's age and reliability have direct relations with routing algorithm. By changing the routing algorithm and controlling source-destination shortest paths selection, the aging impact on NoCs can be mitigated.

1.2 Dissertation Objectives and Contributions

The elevated temperature in 3D systems can be solved by *Thermal Through Silicon Vias* (TTSVs) which provide a promising potential solution to thermal management in 3D ICs by lowering the thermal resistance of their dies and facilitating heat transfer across them. Past research has either overestimated or underestimated the effects of TTSVs as a consequence of the lack of detailed 3D IC models or system-level simulations. In this dissertation, we propose a simulation flow to accurately simulate TTSV effects on 3D ICs. The temperature profiles of 2D and 3D modified Nehalem x86 processor are investigated, then TTSVs are later placed close to hot spot regions to facilitate heat dissipation. First, we adopt benchmarks from Splash-2 running on a full system mode of gem5 simulator while McPAT is employed to generate the corresponding power consumption. Then, the power traces are fed to HotSpot for thermal simulation and temperature profiling. By using detailed 3D thermal model along with full system and validated thermal simulators, our results show accurate thermal analysis of 3D ICs. In addition, we found that the peak temperature of 3D Nehalem is reduced with a small area overhead. Furthermore, we present a hierarchical approach to optimize the floorplan of a 3D Nehalem-based multicore processor via *Simulated Annealing* (SA) with respect to the area, temperature, and wirelength. Our findings show that an increase in the TTSV area generally accompanies a decrease in peak temperature, but the wirelength depends on the TTSV arrangement, which is uniquely optimized for each case of the allowed TTSV area.

On the other hand, aging is an emerging reliability concern, which degrades the performance of systems and causes timing failure eventually. *Bias-Temperature-Instability* (BTI) and *Hot-Carrier-Injection* (HCI) are the dominant aging mechanisms, which escalate in higher temperature and stress (i.e. usage). Previous mitigation techniques to reduce aging phenomena effects on NoC system are either offline, while aging is strictly influenced by runtime operating conditions, or impose significant overheads to the system. For this purpose, this dissertation also presents an online monitoring method through a *Centralized Aging Table* (CAT) for routers in NoCs. Router's capacity in flits, which are the main stimuli in routers, is predictable and limited for a given period of time. Consequently, stress rate and temperature will be in the predictable ranges, as well. Our methodology uses CAT which is populated by values that represent aging degradation for each different pairs of stress and temperature ranges during a given period of time. Furthermore, utilizing CAT, we propose an online adaptive aging-aware routing algorithm in order to avoid highly aged routers which eventually leads to age balancing between routers. Additionally, our proposed routing algorithm reduces the maximum age of routers by changing the shortest paths between source-destination pairs adaptively, considering routers' ages across them in each given period of time. We also extend this idea to 3D by proposing AROMa, which is an aging-aware deadlock-free adaptive routing algorithm integrated with a novel online aging monitoring system for 3D NoCs. The monitoring system in AROMa exploits *Distributed-Centralized-Aging-Table* (D-CAT) to obtain routers' aging rates for each layer of 3D NoCs periodically. Consequently, AROMa swaps between different k-best source-destination shortest paths periodically to avoid highly aged routers, force them in the recovery phase of BTI, and accordingly balance aging in the network. Our results demonstrate that online routing algorithm and monitoring methodology, CAT, improves delay degradation of maximum aged router and aging imbalance while ensuring that the impact of our proposed methodology on network latency, *Energy-Delay-Product* (EDP) and link utilization is negligible. Moreover, our analysis using gem5 full system mode for PARSEC and SPLASH-2 benchmark suites concludes that AROMa outperforms state-of-the-art works while improving age imbalance and router maximum age with negligible overheads.

In Chapter 2, we propose a simulation flow which combines the system-level modeling and detailed thermal simulation to provide an accurate description of the performance enhancement and thermal issues in 2D and 3D ICs. Chapter 2 contributions can be summarized as follows:

- We conduct a full system level simulation of 3D ICs using gem5 [22] for Splash-2 [134] benchmarks to extract and evaluate all the defined component activities. The components' activities output along with the system configuration are fed to McPAT [88] to obtain power consumptions for all components and units.

- Accurate temperature distributions of a 2D and 3D x86 processor with Nehalem floorplan [84] are simulated using the HotSpot tool [70]. The temperature profiles of 3D Alpha 21264 ev6 chip obtained from HotSpot tool and Ansys simulator are compared as a case study. To our knowledge, this is the first time that a comparison of thermal results of 3D ICs between the HotSpot tool and finite element simulator has been reported.

- A detailed thermal model of 3D ICs including TIM, TSVs, BEOL layers, and D2D layers are presented, both primary heat path upwards through the heat sink and secondary heat path downwards through package substrate are considered. A simple but effective TTSVs placement method that assigns TTSVs close to the hot spot regions is proposed and tested. The simulation result shows that putting TTSVs close to the hot spot area can cool the 3D IC significantly with a small chip area overhead.

In Chapter 3, multithreaded application benchmarks are applied to the system-level simulator and the TTSV arrangement optimization occurs early in the design phase. To be more accurate with thermal modeling, we apply the *Effective Medium Theory* (EMT) to capture direction-dependent effective thermal conductivities of TSV units. This chapter presents a hierarchical floorplanning method with a TTSV arrangement optimization algorithm for 3D multicore processors. The contributions of Chapter 3 are summarized as follows:

1. We conduct full system-level simulations using gem5 [22] to extract the component activities of Splash-2 benchmarks [135, 134] and calculate the power density of each IC block of a 3D Nehalem-based multicore processor (referred to as 3D Nehalem in

the rest of the chapter) [84] using McPAT (Multicore, Power, Area, and Timing) [88]. We demonstrate that a critical benchmark (BARNES) can be selected to guide the floorplan optimization process.

2. We develop analytical models based on the EMT to capture lateral and vertical effective thermal conductivities of TSV unit cell. The impact of the liner layer is also considered and the results are validated by the *Finite-Element Method* (FEM).

3. We provide a TTSV placement algorithm based on SA, which optimizes the temperature, wirelength, and area of the floorplan. TTSV blocks with the direction-dependent thermal conductivities are considered during the optimization process.

4. We present a hierarchical approach to generate the floorplan of multicore 3D ICs, which utilizes an optimized single-core to complete the core layer floorplan of the multicore 3D IC based on a symmetric operation. The impacts of TTSV dimensions and area overheads on the peak temperature and wirelength of 3D ICs are discussed in detail.

In Chapter 4, we propose a methodology based on a *Centralized Aging Table* (CAT). CAT is populated by the amount of aging degradation for different ranges of fl and rs in a router from zero up to the router capacity. This makes CAT independent from the running application. CAT, which is stored in one of the cores, can be accessed by all routers through the NoC in order to accumulate their current age to the pre-evaluated respective aging degradation. To compute fl and rs, a counter and a timer is embedded into each router. Our routing algorithm finds k-best shortest paths and selects between them periodically based on the impact of aging on routers (using CAT) to reduce maximum aged router and balance the age between routers. Since aging mechanisms impacts critical path's delay gradually, we update the routing tables in periodic time (P) (e.g. each week).

In Chapter 5, the work in Chapter 4 is extended to 3D NoCs and its main contributions are summarized as follows:

- We formulate aging effects cause by HCI and BTI for 3D NoCs using our proposed online monitoring distributed centralized aging table (D-CAT) in AROMa. D-CAT is able to quantify the gap imposed by temperature change in different perpendicular distances of layers from the heat sink and system's conditions expressed by stress in 3D NoC. Using D-CAT, routers are able to keep track of their ages at each determined time ϵ.

- We propose ***A**ging-aware **R**outing algorithm and **O**nline **M**onitoring* (AROMa), an online adaptive aging-aware, deadlock free routing algorithm, and online monitoring system for 3D NoCs. AROMa chooses one of k-best shortest paths between each source-destination pairs, which has least aged routers by avoiding the maximum aged ones. This adaptivity happens at each period of time $P = n \times \epsilon$.

- We proved AROMa is a deadlock-free technique.

- We implemented AROMa using gem5 full system mode and compare it to ***OF**fline budgeting for adaptive **A**ging-aware **R**outing* (OFAR) and **N**on-**A**ging a**W**are routing (NAW) for both 2D and 3D NoCs.

1.3 Dissertation Organization

In Chapter 2, we analyze 3D *Integrated Circuits* (ICs) with and without the inclusion of TTSVs. A system level approach is detailed that demonstrates the benefit of adding TTSV to the 3D structure in term of heat dissipation. First, we introduce our approach and investigate its current related research. Then the composition of thermal models that have been employed in our analysis is discussed including TTSVs placement in 3D ICs, BEOL and D2D Layers. Next, Hotspot, which is the tool that we used in our thermal analysis and temperature profiling, is validated against the state of the art tools as well as theoretical thermal models. To this end, Hotspot's thermal analysis on a case study processor, namely ev6, is conducted. In addition, we show the simulation model and describe our Nehalem multicore floorplan components which is used in our experimentation. After that, we demonstrate the effect of adding TTSVs on heat dissipation which leads to significant reduction of the elevated temperature of the entire 3D IC stack. Finally, we discuss our finding regarding the thermal management of 3D Nehalem processor using TTSVs then conclude the chapter.

In Chapter 3, thermal TSV optimization and hierarchical floorplanning for 3D ICs is investigated. It is different from Chapter 2 because the addition of the TTSVs to the 3D IC is optimized using *Simulated Annealing* (SA). After introducing the idea and discussing its associated related work, thermal characterization and modeling of 3D Nehalem is explained in order to optimize its floorplan with the presence of TTSV in term of area, wirelength and temperature. For this purpose, we first give a formal representation of the analytical models of a TSV unit cell. Then, we explain our framework and formulate the problem of TTSVs placement. Next, our method of TTSVs optimization and hierarchical floorplanning is outlined. Finally, the impacts of TTSVs on the peak temperature reduction of the 3D Nehalem and its associated area and wirelength overhead is thoroughly explained with conclusive remarks.

In Chapter 4, online monitoring and adaptive routing for aging mitigation in 2D NoCs are illustrated. First, we introduce the delay degradation effect on NoCs and review the works that have been performed in this subject. Next, we explain our online method for observing the aging problem in NoC routers using *Centralized Aging Tables* (CAT), which directly affect their associated cores themselves. After that, our aging aware routing algorithm is demonstrated in details. Eventually, the experimental setup and our findings are discussed with overhead analysis, then the chapter is concluded.

In Chapter 5, we present our A*ging-aware deadlock-free Adaptive* R*outing algorithm and* O*nline* M*onitoring* (AROMA) in 3D NoCs. First, we specifically discuss the most recent studies related to 3D NoC aging including congestion aware, fault tolerant and aging aware adaptive routings. Then, 3D NoCs brief background is provided. Next, aging-induced delay degradation background with its associated BTI and HCI aging impact as well as the joint impact of them. After that, our problem formulation is outlined and its solution by online aging monitoring in 3D NoC using Distributed, Centralized Aging Tables, or D-CAT. We develop and implement an algorithm for adaptive aging-aware routing algorithm based on online monitoring, and we show the simulation result of this algorithm. Also the notion of delay degradation effect reduction analysis and the results are discussed. Finally, the chapter is concluded with our overall findings.

In Chapter 6, we present the conclusion of the entire dissertation and an outlook for the potential improvements and future research in this field.

Chapter 2

System Level Analysis of 3D Integrated Circuits with Thermal TSVs

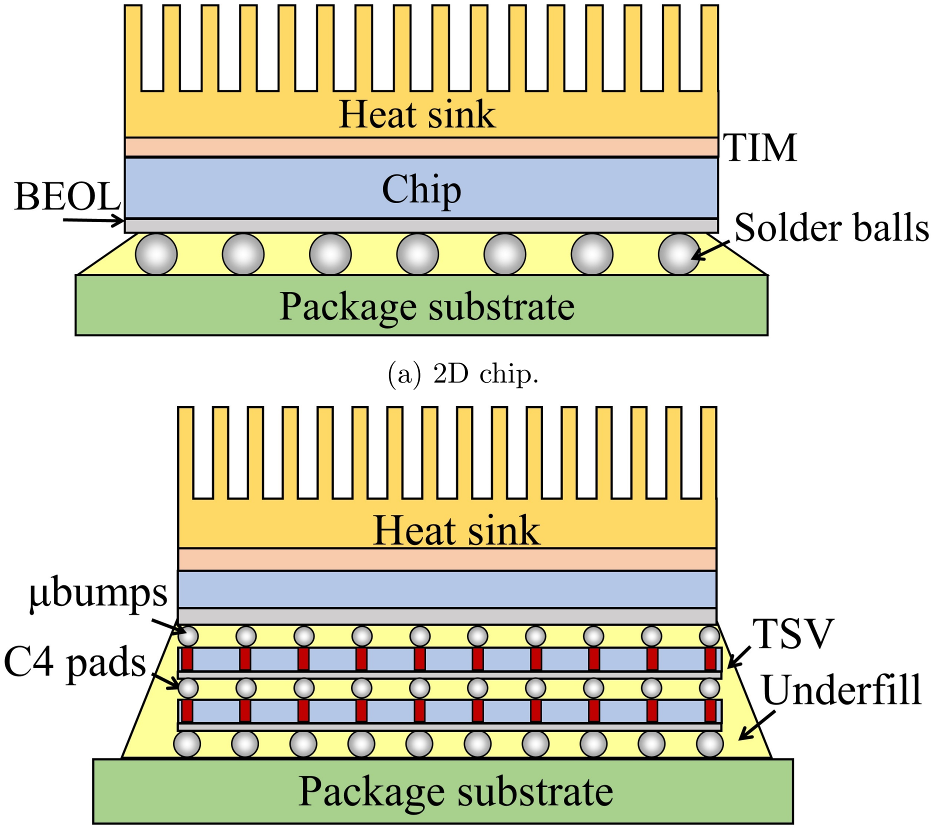
3D stacking of ICs provides significant advantages in saving device footprints, improving power management, and continuing performance enhancement, particularly for many-core systems. However, the stacked structure makes the heat dissipation a challenging issue. While TTSV is a promising way of lowering the thermal resistance of dies, past research has either overestimated or underestimated the effects of TTSVs as a consequence of the lack of detailed 3D IC models or system-level simulations. Here, we propose a simulation flow to accurately simulate TTSV effects on 3D ICs. We adopt benchmarks from Splash-2 running on a full system mode of the gem5 simulator, which generates all the system component activities. McPAT is used to generate the corresponding power consumption and the power traces are fed to HotSpot for thermal simulation. The temperature profiles of 2D and 3D Nehalem like x86 processor are compared. TTSVs are later placed close to hot spot regions to facilitate heat dissipation, the peak temperature of 3D Nehalem is reduced by 5-25% with a small area overhead of 6%. By using a detailed 3D thermal model, full system simulation, and a validated thermal simulator, our results show accurate thermal analysis of 3D ICs.

The rest of this chapter is organized as follows: an overview is given in the introducing Section 2.1. In Section 2.2, we discuss related work of 3D ICs in terms of TTSV cooling solution and simulation tools. In Section 2.3, we propose a detailed thermal model of 3D ICs. In Section 2.4, we present a case study showing the overall chip temperature changes of 2D to 3D ev6. The case study also shows a comparison between two major thermal simulation tools and the cooling effect of TTSVs on 3D ev6. Section 2.5 discusses the simulation framework and 2D and 3D Nehalem floorplan. Then, the simulation setup and results are discussed in Section 2.6. Finally, we conclude the chapter in Section 2.7.

2.1 Introduction

It is well known that Moore's law is approaching its physical limitations, where the continuous shrinking of transistor size and increasing of its density on a microprocessor is becoming intractable. 3D ICs may offer one of the most practical solutions to extend Moore's law by expanding the architecture in the third dimension to reduce the footprint and short the metal wires [24, 85, 81]. Stacking of the chips is made possible by the D2D connections and TSVs. The D2D layers are made of microbumps (or μbumps) and underfill, which exist between dies for physical connections. TSVs run across the bulk silicon die, which serve as electrical connections between active layers [27]. The move from 2D to 3D stacking for SoC and NoC introduces a methodology for integrating a very high number of logic in a single die. The achievable performance benefits arising out of adopting 3D stacking in these systems include performance gain, functionality, and packaging density, especially in terms of throughput, latency, energy dissipation, and wiring area overhead [52]. The work in [110] also showed that 3D NoCs achieve better average performance than 2D NoCs through both analysis and simulation. 3D ICs attribute clear performance improvement as compared to 2D ICs, however, the thermal issues have become a severe problem as a direct result of the high power density and stacked structures. The power consumption of a 3D IC is expected to decrease because of the shorter interconnects, but the power density increases caused by the high number of active devices per unit volume [33]. The stacked dies in 3D ICs are also detrimental to thermal management. As shown in Figure 2.1, for 2D ICs, the heat generated by the active circuits can be easily accessed and dissipated through the bulk silicon substrate and TIM to the heat sink structures, which undergoes convection heat transfer to 25 °C ambient air. Only a small fraction of the heat flows downwards through the BEOL wiring layers and the solider balls into the package substrate because of this path's high thermal resistance. For 3D ICs, there is no direct access to the stacked chips nor a method of dissipating heat laterally from the stacked dies, both the BEOL wiring layers and the μbump layer between the stacked chips are now in the thermal path [36]. The increased power density and the various low thermal conductivity interlayer materials have made it significantly difficult to maintain all the dies at an acceptable junction temperature. The challenges in thermal management can be even higher due to the non-uniform power dissipation, which leads to the formation of local hot spots that cannot be easily eliminated by conventional heat sink structures [15, 112].

High chip temperature increases the risk of damaging devices and interconnects. Current cooling schemes of 3D ICs can be classified into two categories: heat sink optimization methods and internal heat distribution optimization methods [62]. Heat sink optimization methods including air cooling by electrical fans or micro-channel cooling at heat sinks, however, these methods cannot provide effective cooling for 3D ICs because the high thermal resistance D2D layers impede the heat dissipation from active circuits to heat sink structures [128]. The internal heat distribution optimization methods are to insert micro-channels or TTSVs inside the silicon chips to facilitate heat transfer from the stacked structures to heat spreaders. Micro-channel cooling scheme is costly in fabrication and dielectric liquids,



(a) 2D chip.

(b) 3D ICs.

Figure 2.1: Schematic diagram of liquid-less chip package

it also creates large obstacles for TSVs [74]. TTSVs, on the other hand, offer clear favorable attributes including solid-state operation and electronic process compatibility [86]. Previous studies have focused on the performance improvement from 2D ICs to 3D ICs and investigated the possible cooling solutions, however, most of the work has failed in accurately describing the temperature profiles of 3D ICs, where either a full system level simulation or detailed thermal models are missing. In this work, we propose a simulation flow which combines the system-level modeling and detailed thermal simulation to provide an accurate description of the performance enhancement and thermal issues in 2D and 3D ICs. A simple but effective TTSVs placement method is presented, the peak temperature of 3D Nehalem can be reduced by 5-25% for different Splash-2 benchmarks with a TTSVs block area overhead of 6%. Following is a summary of the contributions of this work:

- We conduct a full system level simulation of 3D ICs using gem5 [22] for Splash-2 [134] benchmarks to extract and evaluate all the defined component activities. The components' activities output along with the system configuration are fed to McPAT [88] to obtain power consumptions for all components and units.

- Accurate temperature distributions of a 2D and 3D x86 processor with Nehalem floorplan [84] are simulated using the HotSpot tool [70]. The temperature profiles of 3D Alpha 21264 ev6 chip obtained from HotSpot tool and Ansys simulator are compared as a case study. To our knowledge, this is the first time that a comparison of thermal results of 3D ICs between the HotSpot tool and finite element simulator has been reported.

- A detailed thermal model of 3D ICs including TIM, TSVs, BEOL layers and D2D layers are presented, both primary heat path upwards through the heat sink and secondary heat path downwards through package substrate are considered. A simple but effective TTSVs placement method that assigns TTSVs close to the hot spot regions is proposed and tested. The simulation result shows that putting TTSVs close to the hot spot area can cool the 3D IC significantly with a small chip area overhead.

2.2 Related Work

This section presents two categories of TTSVs placement methods and covers the widely used system-level and thermal simulation tools including the consequences and the advantages of adding them.

2.2.1 Thermal TSV Placement in 3D ICs.

TTSVs cooling solution is at the heart of thermal management of 3D ICs, the density and placement of TTSVs have a substantial impact on the performance, reliability, and temperature of 3D ICs [86]. The placement methods of TTSVs in 3D ICs can be classified into two categories.

TTSV Placement Using Optimization Algorithms

One category of thermal aware TSV placement techniques relies on specific algorithms and weight functions. Budhathoki et al. [26] demonstrated a localized TSV placement algorithm that used a greedy approach to place TTSV in the passive substrate and bonding layer until the maximum temperature of each grid cell is below the target temperature. This method reduced the maximum chip temperature at a reasonable TTSV density, but they did not account the secondary heat path through solder bumps and the package substrate. Chen et al. [29] described a finite element code coupled with a sequential quadratic programming algorithm to determine the optimal thermal design of TTSVs, but they lacked full-system simulation and the topology optimization of TTSVs for different benchmarks. As the circuits and power consumption increase, efficient algorithms are required to accurately determine the distribution of TTSVs to cool 3D ICs, but an efficient, systematic and automated thermal design tool that facilitates the combined thermal design including detailed thermal models and full system simulations is currently unavailable.

Thermal TSV Placement to Specific Locations and Areas

Goplen and Sapatnekar [62] reported that by assigning the TTSVs to specific areas of a 3D IC and adjusting their effective thermal conductivity, the same temperature reduction can be achieved with less than 50% TSV area compared to a uniform TTSVs placement. Agrawal et al. [4] showed that by aligning and shorting dummy D2D μbumps with TTSVs, it enabled an average increase in processor frequency of 720 MHz with an area overhead of 0.81% without exceeding acceptable temperatures. Adding TTSVs to specific locations is easy to apply through all the silicon layers without breaking the logics in core layers and cache layers, and where to add the TTSVs can reduce the temperature more efficiently becomes an important question. The peak temperature of 3D ICs is usually determined by the blocks that have the highest power density, in this study, we first run Splash-2 benchmarks to find the regions that have the maximum temperature, after that, we assign TTSVs blocks close to the hot spot regions through all the layers in 3D ICs to facilitate heat dissipation from the stacked chips to the heat sink structures.

System Level Simulation Tools for Die Temperature Estimation

The following subsections overview the state-of-the-art system level simulators in architecture research as well as thermal simulators.

System Level Simulator

There are several system level simulators that generate cores activities. They differ in many aspects such as their scope, execution accuracy, synchronization techniques for running concurrent simulations, support for full system simulation, and whether they are trace or execution-driven simulation. Table 2.1 summarizes six of the most popular simulators and their properties. The scope of the simulator can be either user level with operating system functionality emulation or system level with the full support of a functioning operating system such as Linux Ubuntu. The accuracy of the simulator can be, cycle by cycle accurate, adopt an emulation technique that reduces the complexity of a rigorous cycle-based simulation, or a hybrid approach. The experimentation parallelism when more than one core is tested in a simulator can be sequential, or parallel with synchronization ways that are either cycle by cycle or intermittently synchronized. In our research, we adopt gem5 which is an event-driven system simulator with a modular platform for computer system architecture research, including system-level architecture as well as processor microarchitecture. Gem5 main advantages are its support for full system mode and its high accuracy.

Table 2.1: System level core simulators

Name	Scope	Accuracy	Parallelism
Gem5 [22]	User/Full system	Cycle accurate	Sequential
Marss [104]	Full system	Hybrid (cycle/emulation)	Sequential
Sniper [28]	Full system	DBT[a] emulation	Loose[b]
Zsim [116]	User	DBT[a] emulation	Loose[b]
Manifold [130]	Full system	Cycle/hybrid/DBT emulation	Conservative[c]/loose
Hornet [111]	User	Cycle accurate	Conservative[c]/loose

[a] DBT stand for dynamic binary translator

[b] loose parallelism is synchronized intermittently (not every cycle)

[c] conservative parallelism reduce the synchronization frequency

Thermal Simulators

Several tools have been used to study the temperature profiles of 3D ICs, Lau and Yue [86] observed the thermal performance of the 3D system in package (SiP) with copper filled TSV based on CFD analysis. Chien et al. [34] used Icepack as the simulation tool to solve the heat conduction problems of 3D SiP by finite volume method. Besides these, HotSpot is a temperature modeling tool that has been widely used in computer architectural studies [4, 26, 95]. In our research, we use the HotSpot's extension as suggested by Meng et al. [95] to account the thermal properties of different layers for modeling the thermal effects of 3D ICs, we further validate the thermal results of HotSpot tool using a finite element simulator

Ansys in the case study.

2.3 Composition of Thermal Models of 3D ICs

In this study, we model a Nehalem like x86 processor. As shown in Figure 2.1b, the package of 3D ICs is composed of many distinct layers, such as heat exchangers, silicon chip, metal layers, and D2D layers. Metallic TSVs are presented in the silicon chip layers and connected to each other by the μbump. The chip stacking can be classified into three configurations: the active layer of the die faces the bulk layer of another (F2B), the active layer of dies facing each other (F2F) and the bulk layer of dies facing each other (B2B) [4]. We assume the F2B configuration in our simulation because its stacking unit process can be repeated multiple times and is favorable for stacking multiple dies.

2.3.1 Through Silicon Vias (TSVs)

TSVs are mainly metallic pillars that completely pass through a silicon die, whose main purpose is to establish electrical connections between stacked dies [77]. There are several fabrication processes for TSVs, such as via-first, via-last and via-middle [32]. A typical diameter of via-first TSVs ranges from 1 μm to 5 μm, whereas that of via-last TSVs ranges from 5 μm to 20 μm [18]. The presence of the TSVs also increases the thermal conductivity of silicon substrate, researchers have proposed to use dummy TSVs which do not transfer any purposeful signals and simply dissipate heat, as opposed to signal conduction, these TSVs are called thermal TSVs [18, 26, 34, 38, 62]. Non-metallic materials like diamond or Boron Arsenide may attribute higher thermal conductivity and provide better cooling performance, but, metallic materials like Cu or W have been widely used for their well-established fabrication process [85]. For metallic TTSVs fabrication, there is a tradeoff between the TSVs density and die thickness. To facilitate the heat transfer, a high TTSVs density is preferred, which needs a thinner die ascribable to the manufacturing constraints. However, a small die thickness reduces the thermal spreading effect and impairs the hot spot heat dissipation [49]. In our simulation, we assume a via-last technique with a TSVs diameter of 20 μm and a pitch size of 40 μm. The main heat transfer path in 3D ICs is in the cross-plane direction, as shown in Figure 2.1, the thermal conductivity of silicon chip with copper TSVs can be obtained using the effective medium theory [100]:

$$k_{eff} = k_{Cu}(\varphi) + k_{Si}(1 - \varphi) \tag{2.1}$$

where k_{eff} is the effective thermal conductivity, k_{si} and k_{cu} is the thermal conductivity of silicon and copper, respectively. φ is the fractional area occupancy of copper TSV in silicon chips. Using the effective thermal conductivity, the temperature profiles of 3D ICs can be obtained with less compactional recourses [34].

2.3.2 BEOL Wiring layers

The BEOL of current high-performance processors contains ten or more wiring levels and up to 10 μm thick with a variety of dielectric materials [47]. It is necessary to accurately characterize the additional thermal resistance from the BEOL. Colgan et al. [37] measured that the unit thermal resistances of thirty-nine different BEOL test sites consisting of four line levels and three via levels were in the range of 0.5 - 5.5 mm^2KW^{-1}, in our simulation, we assume the thickness of BEOL is 2 μm and the equivalent thermal conductivity is 1 $Wm^{-1}K^{-1}$.

2.3.3 Die-to-Die (D2D) Layers

The D2D layers include underfill/air, SiO_2,SiN and μbumps. μbumps provide the interconnection between stacked dies and can be classified as electrical or dummy μbumps. Electrical μbumps provide signal connections between the active layers while dummy μbumps are used for mechanical and thermal transport [36]. Previous research underestimated the thermal resistance of D2D layers by assuming a high conductivity or a small thickness [62, 91]. Recent experimental data [36] showed that for 25 μm diameter Pb-free solder μbumps with pitch sizes of 50 - 100 μm, the average thermal resistance with underfill was ranged 8 - 20 mm^2KW^{-1}, making it the true thermal bottleneck in the 3D ICs. The complexity of D2D composition makes it hard to simulate using a detailed model. In our simulation, we assume the thickness of the D2D layer is 20 μm and use an equivalent thermal conductivity of 2.5 $Wm^{-1}K^{-1}$. Table 2.2 shows a summary of the dimensions and thermal conductivities of each layer in the stacked chips.

Table 2.2: Dimensions and thermal parameters

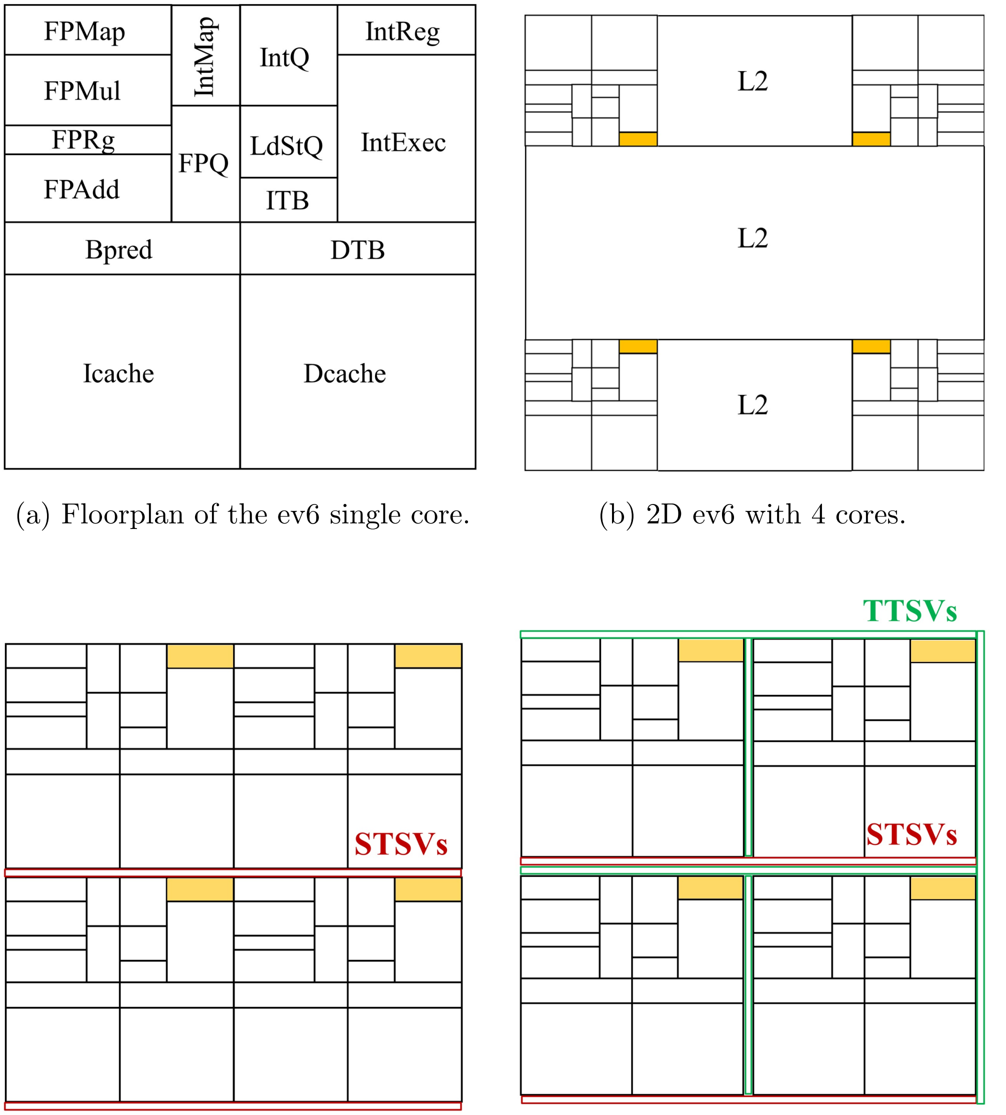
Layer	Dimensions (μm)	Thermal conductivity (W/mK)
TIM	20	4
Silicon chip	100	120
BEOL	2	1
D2D layer	20	2.5
Silicon chip with TSVs	100	100 (Si); 400 (TSV); (170 TTSV bus)

2.4 Hotspot Validation and Thermal Analysis on a Case Study Processor- ev6

In this section, we show the difference of the temperature profiles of the ev6 processor of 2D and 3D cases using HotSpot 6.0 [139], in addition, we compare the thermal results of 3D ev6 between Ansys and HotSpot for validation purpose and provide preliminary discussions of thermal TSVs.

2.4.1 Floorplan of 2D and 3D ev6

The cross-sectional views of 2D and 3D chip packages are shown in Figure 2.1. The detailed information about material properties can be found in Table 2.2. The floorplan of the ev6 core is shown in Figure 2.2a and the floorplan of 2D ev6 is shown in Figure 2.2b, for the 3D case, four ev6 cores are placed in one layer (Figure 2.2c) connecting to the shared two L2 caches in lower layers using STSVs. The core layer is placed close to the heat sink for better heat dissipation. We model the chip with *gcc* benchmark, and the IntReg block (highlighted in yellow in the floorplans of 2D and 3D ev6 in Figure 2.2 is expected to have the maximum temperature because of its highest power density. For the thermal management of 3D ev6, we assign TTSVs close to the IntReg block with the same dimension as the STSVs (Figure 2.2d). The TSV diameter is 20 μm and the volume ratio is 24%, there are 3400 STSVs for Figure 2.2c and 6800 TTSVs for Figure 2.2d. The total area of L2 cache and power are the same for 2D and 3D ev6 for a fair comparison, the temperature profiles of different architectures are simulated with HotSpot tool and validated by the finite element simulator Ansys. In the Ansys simulation, the whole structure of 3D ev6 including a heat sink, a heat spreader, TIM layers, and silicon chips is built using the same dimensions as in HotSpot. Convection heat transfer is applied on the top surface of the heat sink and heat flux from *gcc* benchmark is applied to the bottom surface of the silicon chip, they are the same as in HotSpot tool. In this section, we are providing a simple case study on ev6 to evaluate the temperature increase from 2D ICs to 3D ICs, the cooling performance of TTSVs and validate HotSpot tool and Ansys. We only consider the TIM layers between silicon dies in the case study, for the simulation of Nehalem, a detailed thermal model contains TIM, BEOL, and D2D layer are used.



(a) Floorplan of the ev6 single core.

(b) 2D ev6 with 4 cores.

(c) 3D ev6 without thermal TSVs core layer floorplan.

(d) 3D ev6 with thermal TSVs core layer floorplan.

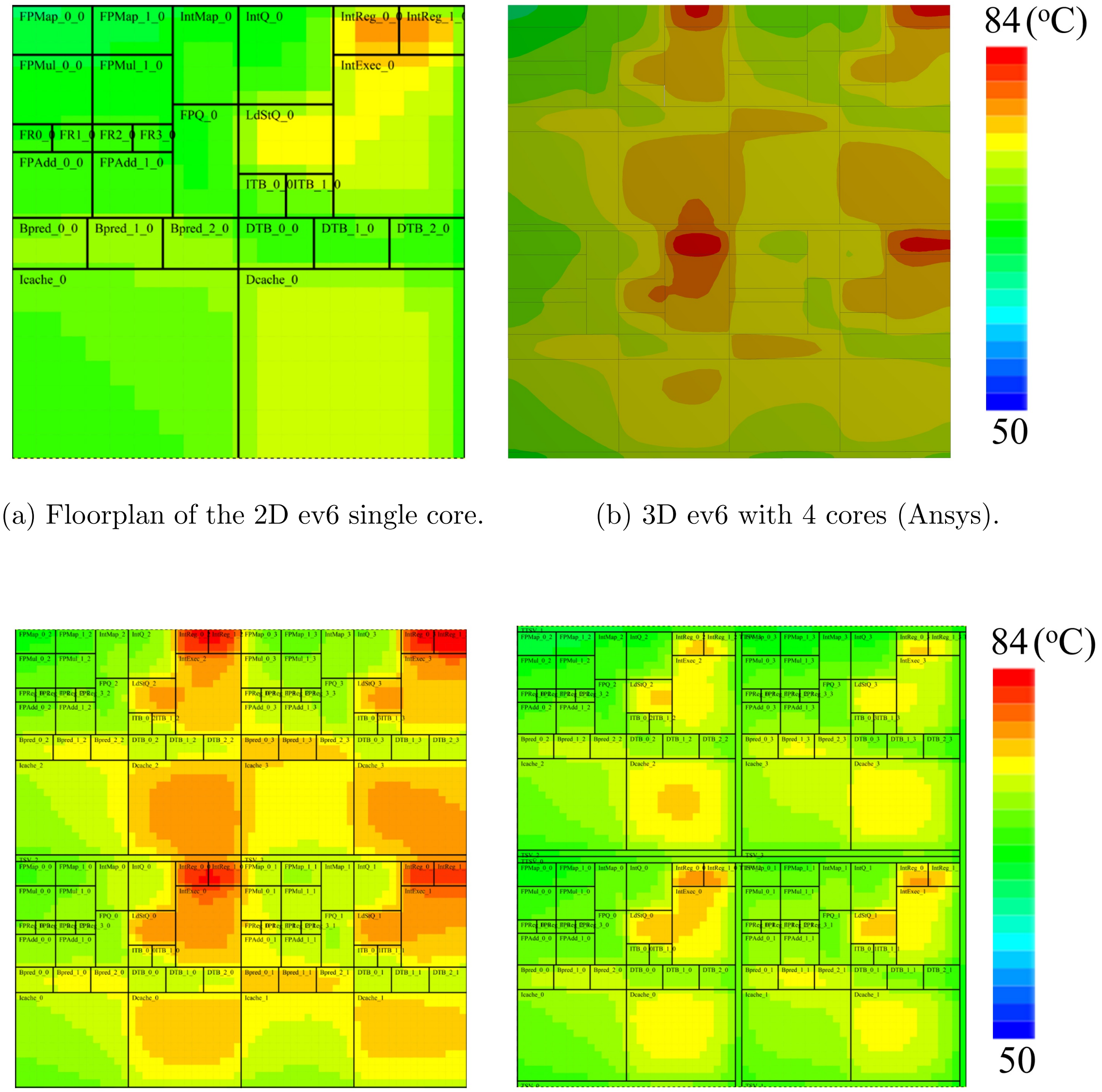
Figure 2.2: Flooorplan of ev6. The hot spot block IntRegs, STSVs and TTSVs are highlighted in yellow, red and green, respectively.

2.4.2 HotSpot Tool Validation Using Ansys and the Thermal Analysis

Figure 2.3 shows the temperature profiles of 2D and 3D ev6. For 2D ev6, the IntReg block has the highest temperature of 76 °C, for 3D ev6, highest temperature occurs at the IntReg blocks of each core and is 10% higher than the 2D ev6 caused by the stacked structure. The 3D ev6 has only one core layer and is close to the heat sink, the peak chip temperature increase is not very significant as compared to 2D ev6. For 3D ICs with multi-core-layers, there will be a substantial temperature increase because of the high thermal resistance D2D layers, which is shown later for the Nehalem floorplan. As shown in Figure 2.3b and Figure 2.3c, the Ansys simulation result is identical to the HotSpot tool, which validates the correctness of these two approaches. The temperature profile of 3D ev6 with thermal TSVs is shown in Figure 2.3d, by placing the TTSVs blocks close to the IntReg along the lateral or vertical direction of the core layer, the TTSVs blocks can penetrate through cache layers without breaking the logics, which further facilitate the heat transfer from inside of the 3D ICs to the heat sink. The temperature of 3D ev6 is reduced by 8% with an area overhead of 4%, this proves our assumption that placing TTSVs close to the hotspot area can effectively cool the 3D ev6.

2.5 Problem Formulation and Our Framework

In this section, we introduce our framework and 2D and 3D Nehalem floorplan. We first use gem5 to get the system-level simulation of Splash-2 benchmarks, then we apply the components activities to McPAT to get the power consumption estimation, finally, we feed the power trace to HotSpot tool to generate the temperature profiles.



(a) Floorplan of the 2D ev6 single core.

(b) 3D ev6 with 4 cores (Ansys).

(c) 3D ev6 with 4 cores (HotSpot).

(d) 3D ev6 with thermal TSVs.

Figure 2.3: The temperature profile of ev6 running *gcc*.

2.5.1 Simulation Model

In this subsection, our adopted system simulator, power estimator and temperature profiler are demonstrated.

System-level Simulation using gem5

We adopt full system level simulation using gem5, which uses specifications including cores, caches, memory, controllers and interconnect component. We simulate Splash-2 benchmarks, which are cross-compiled for the appropriate *Instruction Set Architecture* (ISA) such as ALPHA (ev6) or x86 (Nehalem). Full system-level simulation is made for each benchmark to extract and evaluate all defined component activities including the number of instructions for both of integer and floating-point units, L1/L2/memory reads and writes with their hit and miss rates, NoC statics and more. The flow of gem5 simulation is shown in Figure 2.4.

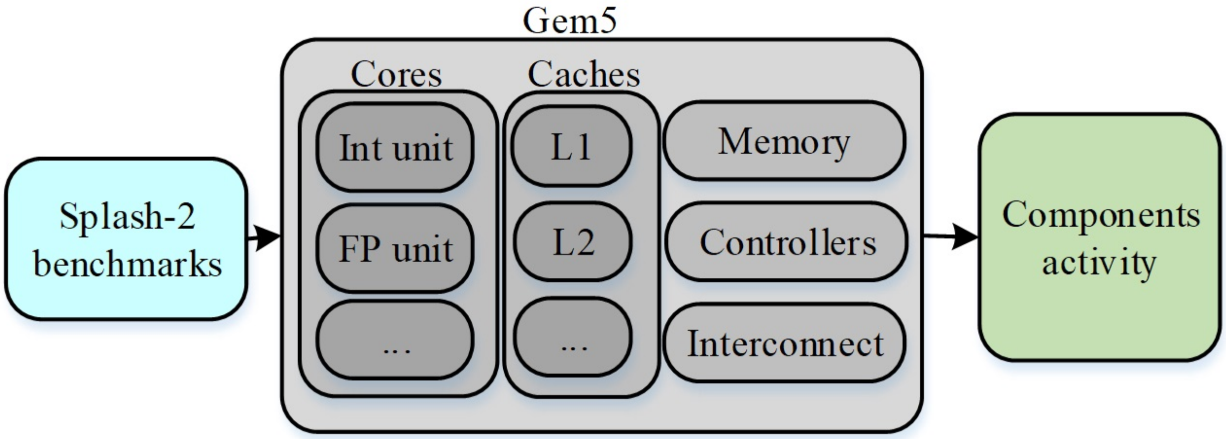


Figure 2.4: Simulation flow of gem5.

Power Consumption Estimation and Thermal Simulation

The components activities output along with system configuration are fed to McPAT, which is an integrated power, area and timing modeling framework for multithreaded, and many-core architectures. All system activities generated by gem5 are given to the McPAT as an input to generate the power traces. The power traces obtained from McPAT are fed to HotSpot for thermal simulation. HotSpot receives the power estimation of all components generated by McPAT, along with a floorplan that is compatible with specifications made in gem5 using thermal properties discussed in Section 2.3. As a result, the temperature of 2D and 3D ICs can be obtained. The whole framework of our simulation is shown in Figure 2.5.

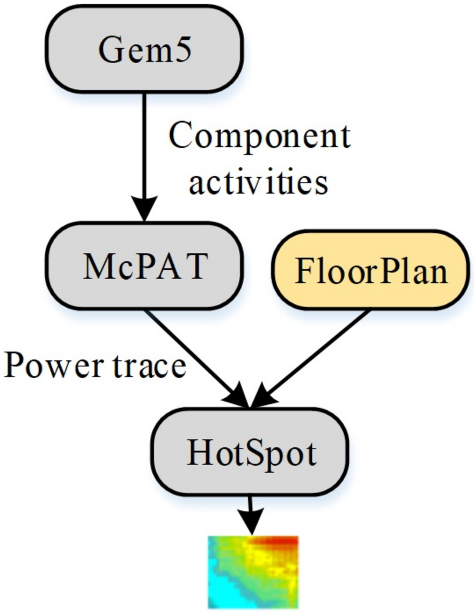


Figure 2.5: Framework of our simulation, from gem5 to HotSpot

2.5.2 Nehalem Floorplan

We adopt x86 ISA out of order Nehalem [84] like processor. The processor is composed of 8 cores with private L1 cache and L2 cache. Threads access data using a shared L3 cache. Figure 2.6a shows core 0 tiles out of the 8 cores which consist of:

1. FP_0: which include the floating-point register file, floating-point units and complex ALUs responsible for divisions and multiplications.
2. Int_0: which include the integer register file, integer units and integer ALUs responsible for integer and logic operations.
3. Icache_0 and Dcache_0: which represent the instruction cache and the load/store data cache, respectively.

4. Itlb_0 and Dtlb_0: which represent the instruction and data translation look-aside buffers, respectively.

5. Ifetch_0: responsible for fetching instructions.

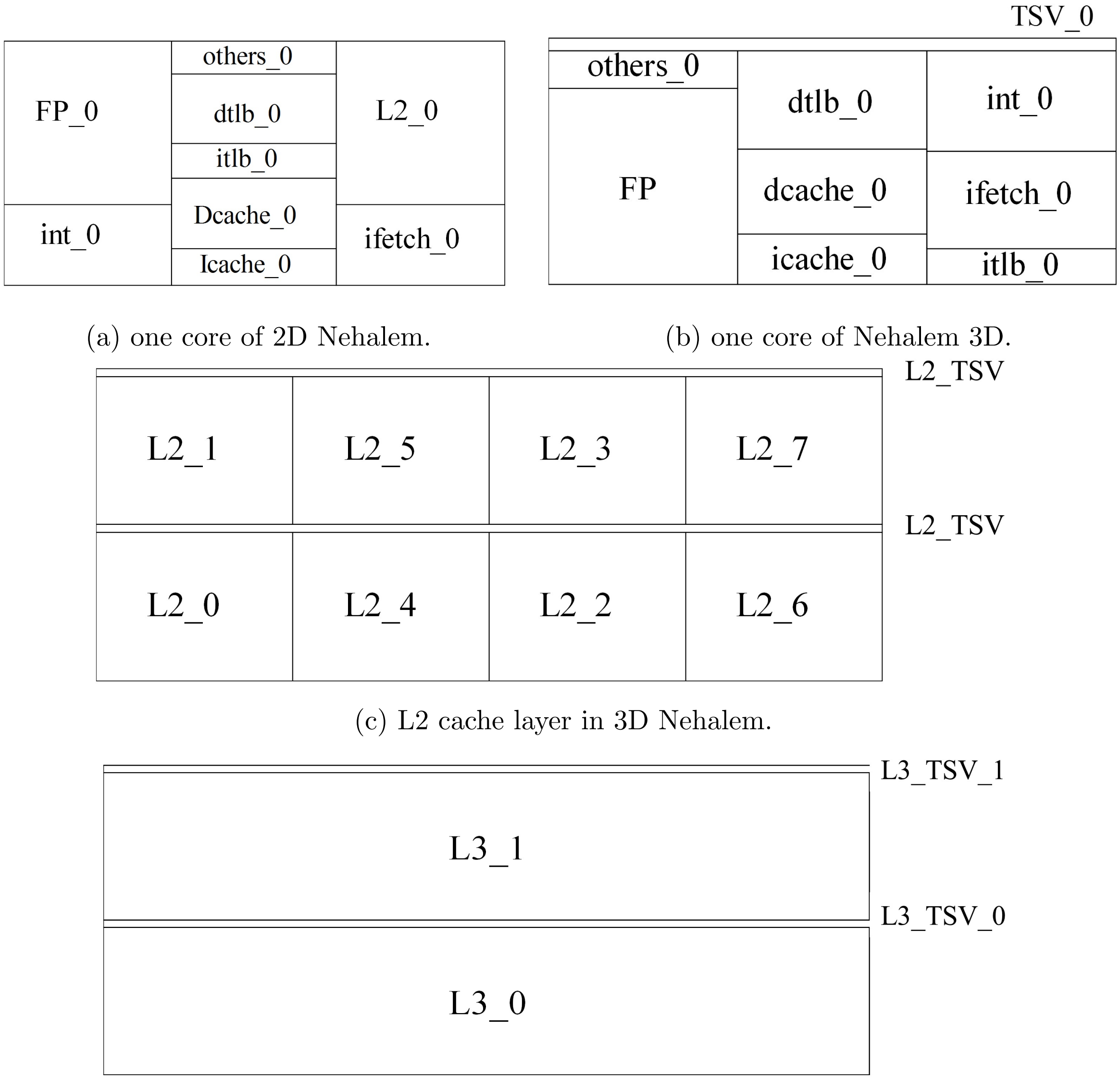
6. L2_0: which represent the L2 private cache.

7. Others_0: include the remaining blocks such as renaming units and *ReOrder Buffer* (ROB).

In the 3D case as it is depicted in Figure 2.10, the stack is composed of two core layers. The first core layer contains four cores from core 0 to core 3. Similarly, the second core layer contains cores from 4 to 7. The next layer contains all L2 caches and the last layer has the L3 cache. Figure 2.6b shows a core tile of the 3D stack that has the same component as the 2D case but missing the L2 cache that is elevated to another layer. The L2 cache layer consists of eight L2 caches for each core and connected to the core layers by using signal TSVs (Figure 2.6c). For geometry matching between layers, each L2 cache size is increased by 50%. Finally, the L3 cache is divided into two banks as illustrated in Figure 2.6d and is connected to the L2 layer using signal TSVs.

2.6 Results and Discussion

In this section, we evaluate our framework and setup of a 2D and 3D Nehalem processors. We first explain the simulation environment setup, then we describe our results for 2D and 3D cases. In addition, we analyze our results in term of temperature and power consumption.



(a) one core of 2D Nehalem.

(b) one core of Nehalem 3D.

(c) L2 cache layer in 3D Nehalem.

(d) L3 cache layer in 3D Nehalem.

Figure 2.6: Nehalem floorplan in 2D and 3D cases.

2.6.1 Simulation Setup

All of our simulations are done in the full system simulation mode using gem5 that runs on Linux operating system. We use 13 benchmarks from Splash-2 with 8 threads each. In order to extract power estimation results for them, we used McPAT for all system component activities which include dynamic power, static leakage power as well as short circuit

Table 2.3: Simulation parameter

Item	Description
Processor	x86 Nehalem like based 2.0 GHz out of order processor with 8 cores.
L1-icache	private, 32KB, 2-way set associative, 64B blocks, 4 cycles latency, with pseudo *Least Recently Used* (LRU) replacement.
L1-dcache	private, 32KB, 2-way set associative, 64B blocks, 4 cycles latency, with pseudo LRU replacement.
L2-cache	private, 256KB, 4-way set associative, 64B blocks, 12 cycles latency, with pseudo LRU replacement.
L3-cache	Shared, 16MB, 8-way set associative, 64B blocks, 30 cycles latency, with pseudo LRU replacement, MOESI cache coherence.
Main memory	1GB DRAM.
Technology process	45 nm.
Threshold Temperature	85°C.

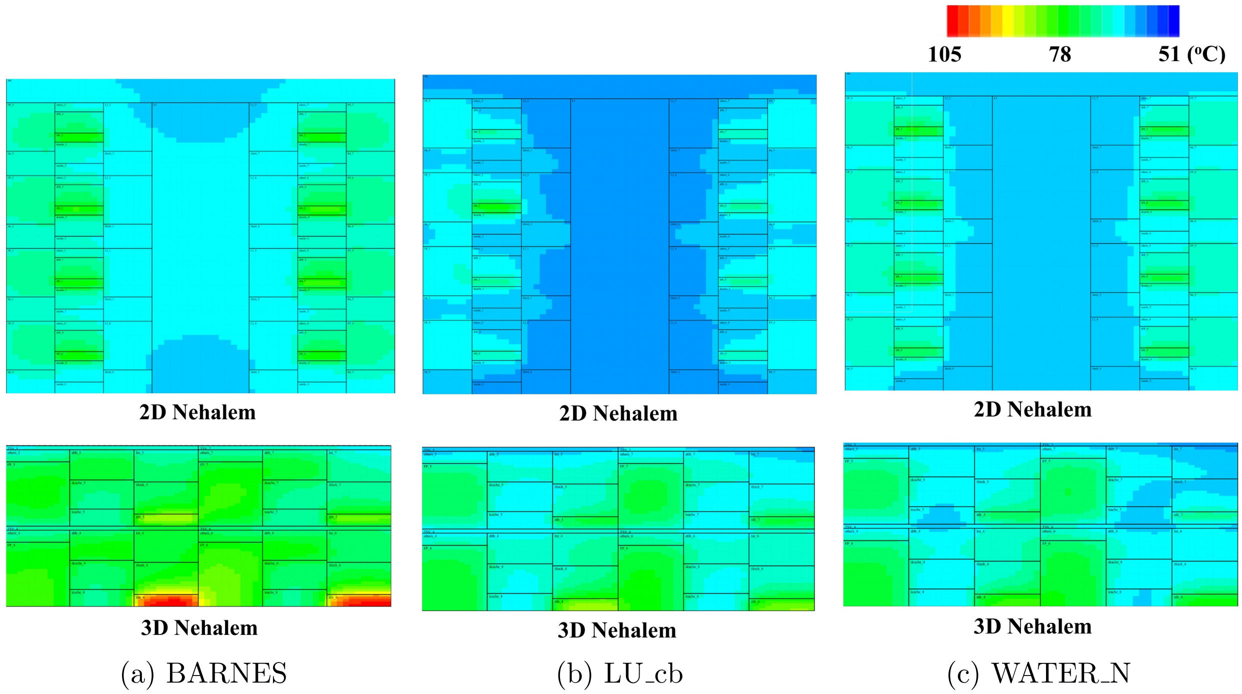
power estimations. HotSpot is used to extract temperature profiles of components for different extracted powers for each multi-threaded benchmark, using 45nm process technology. Splash-2 benchmarks are adopted for our experiments. Each experiment runs with 8 cores. Each core has one L1 instruction cache, one L1 data cache, and one private L2 cache with sizes of 32KB, 32KB, and 256 MB, respectively. A shared 16MB L3 cache is shared between the 8 cores. In all our simulations, the clock frequency is equal to 2 GHz which means the critical path will be 0.5 ns. The temperature threshold is assumed to be 85°C. The rest of the simulation setup is listed in Table 2.3.

2.6.2 Simulation, result and discussion of 2D and 3D Nehalem

In this subsection, we show the temperature profile of 2D and 3D Nehalem and analyze the performance gain and power consumption advantage for using the 3D structure.

Temperature profiles of 2D and 3D Nehalem

Although the 3D Nehalem has the advantage of low power consumption, reduced signal delay, and small footprint, the thermal issues in 3D Nehalem limit its performance and application. The temperature profiles of 2D and 3D Nehalem running Splash-2 benchmarks are obtained using HotSpot tool. The peak temperature varies because of the power density and active regions, among all the benchmarks, BARNES, LU_cb and WATER_N are the three benchmarks that show the top three highest temperatures. Figure 2.7 shows the temperature profiles of core layer Nehalem running these three benchmarks, although the peak temperature is different in 2D and 3D Nehalem, they do have the same hot spot regions. The threshold temperature of 3D ICs to ensure the reliable operation is 85°C [95], as shown in Figure 2.11, there are five benchmarks that exceed and two very close to the threshold temperature. An effective cooling solution is required to cool the processor under the threshold temperature to take advantage of other benefits of 3D ICs.



(a) BARNES (b) LU_cb (c) WATER_N

Figure 2.7: Temperature profiles of 2D and 3D Nehalem running applications.

Performance and Power Consumption

Since wire latency is reduced in 3D stacks, the performance is greatly enhanced as a consequence of short distances in the vertical direction [107, 109]. Figure 2.8 shows the performance gain of the 3D Nehalem over the 2D case in term of execution time. The arithmetic average gain is approximately 20% and the geometric mean is about 14% across 13 benchmarks of Splash-2 suites. The performance is the same for the 3D stacks with or without TTSV or for the worst scenario has a very negligible effect. This occurs because of the lack of communication directly between cores in the same layers. Since all inter-communications occur through the L3 cache, they have insignificant effects on performance. In addition, the power consumption of the 3D case is smaller than the 2D case as shown in Figure 2.9. The average and the geometric mean reduction in power consumption of the 3D case compared to the 2D case is approximately 7%. The reason behind it is that shorter wires also reduce power consumption by producing less parasitic capacitance which results in power consumption reduction.

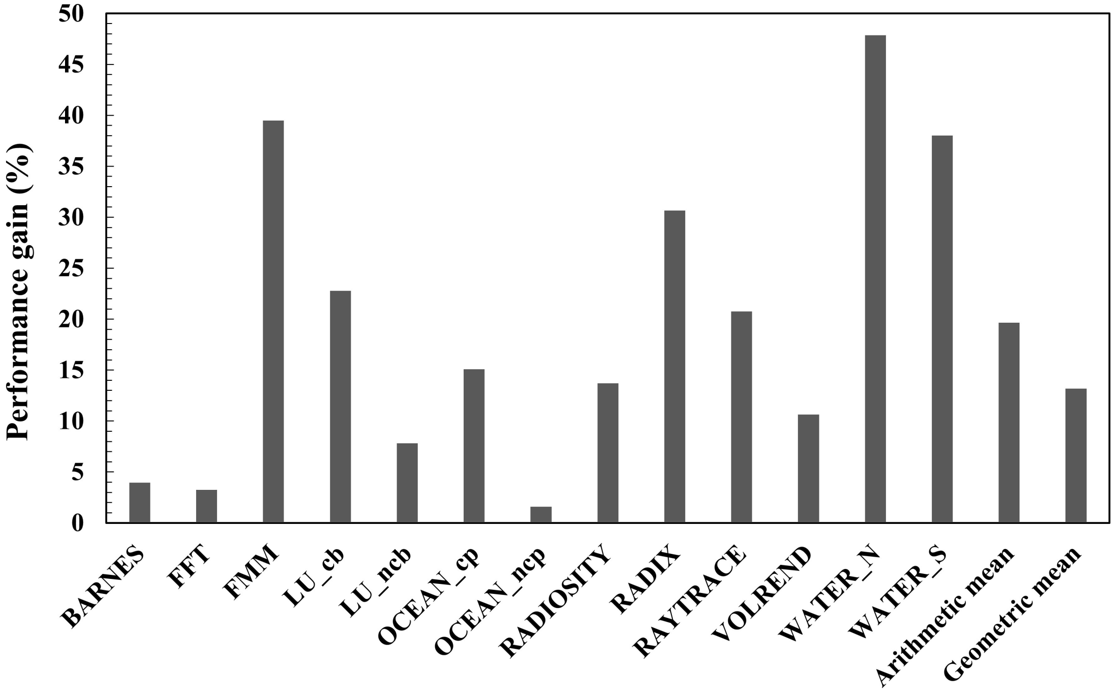


Figure 2.8: Performance of 3D Nehalem

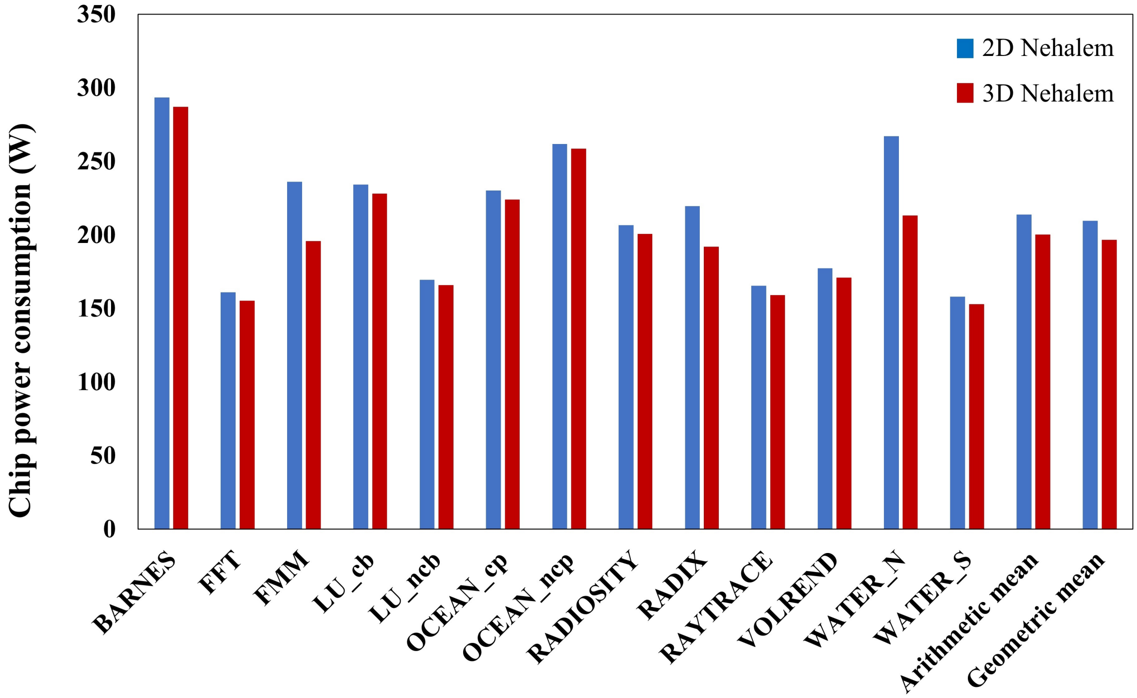


Figure 2.9: Power consumption of 2D and 3D Nehalem.

2.6.3 Thermal management of 3D Nehalem processor using TTSVs

In this subsection, the area overhead and the placement of TTSVs are studied. Then the impact of these TTSVs on temperature reduction is explained. After that, the correlation between chip frequency alternation and the temperature elevation is discussed.

TTSVs Placement and Area Overheads

For these benchmarks, the hot spot regions are *itlb* (highlighted in yellow in Figure 2.10); it is intuitive to place TTSVs close to *itlb* blocks to introduce extra heat transfer paths from the stacked chips to the heat sink or package substrate. The floorplan of 3D Nehalem with TTSVs is shown in Figure 2.10, the TTSVs are highlighted with the green blocks with a total area of 10 mm^2, resulting in 6% area overhead. The diameter and pitch size of TTSVs are assumed to be 20 μm and 40 μm, respectively. Based on the TTSV parameters, the area of one TTSV unit cell is $1.6 \times 10^{-3} mm^2$ and there are 6250 TTSVs in total. TTSVs usually do not have any energy overhead and our placement method can also avoid routing congestion in the core and cache layers.

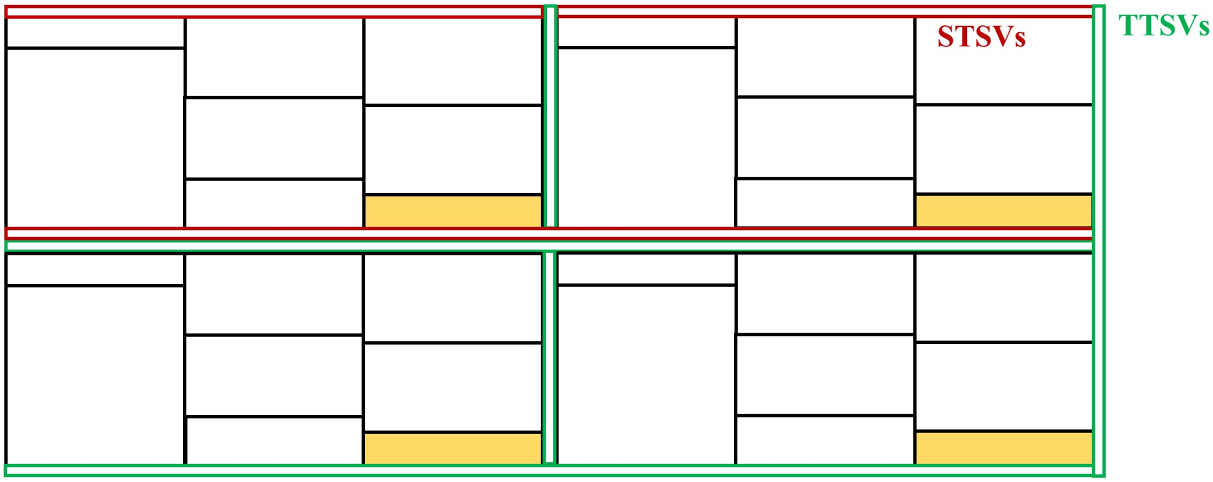


Figure 2.10: Floorplan of 3D Nehalem with TTSVs, the *itlb* block, STSVs, and TTSVs are highlighted in yellow, red and green, respectively

Impact of TTSVs on 3D Nehalem temperature

The temperature comparison of 3D Nehalem without/with TTSVs running other benchmarks is shown in Figure 2.11. We can see that TTSVs can effectively reduce the hot spot temperature of 3D Nehalem to the threshold temperature, even close to the temperature of the 2D case for all benchmarks. Since the TTSVs are dummy blocks, the performance and power consumption remain the same as 3D Nehalem. The thermal simulation shows that by applying the TTSVs close to the hot spot regions all core and cache layers are penetrated and the temperature is reduced consequently. As an example, Figure 2.12 shows temperature profiles of 3D x86 without/with TTSVs running BARNES benchmark (BARNES benchmark has the highest hot spot temperature). The hot spot temperature and area in the core layers are highly suppressed attributable to the TTSVs, the temperature for cache layer and heat sink structures is also reduced which proves that TTSVs can facilitate heat transfer, even in stacked structures and the maximum temperature is reduced more by than 25%.

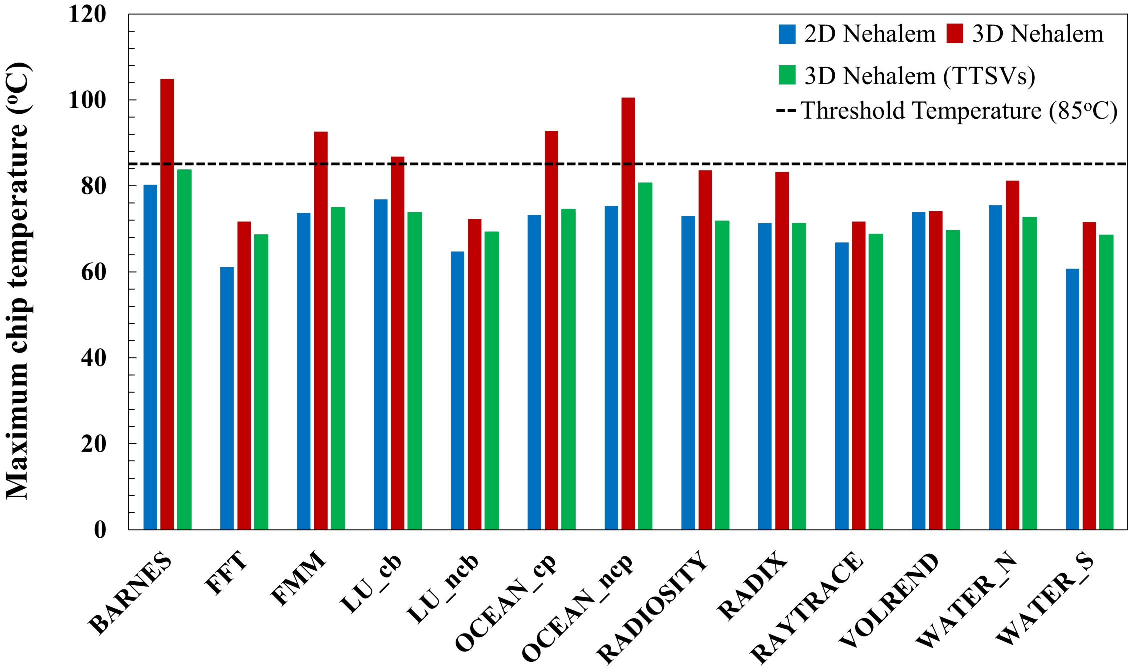
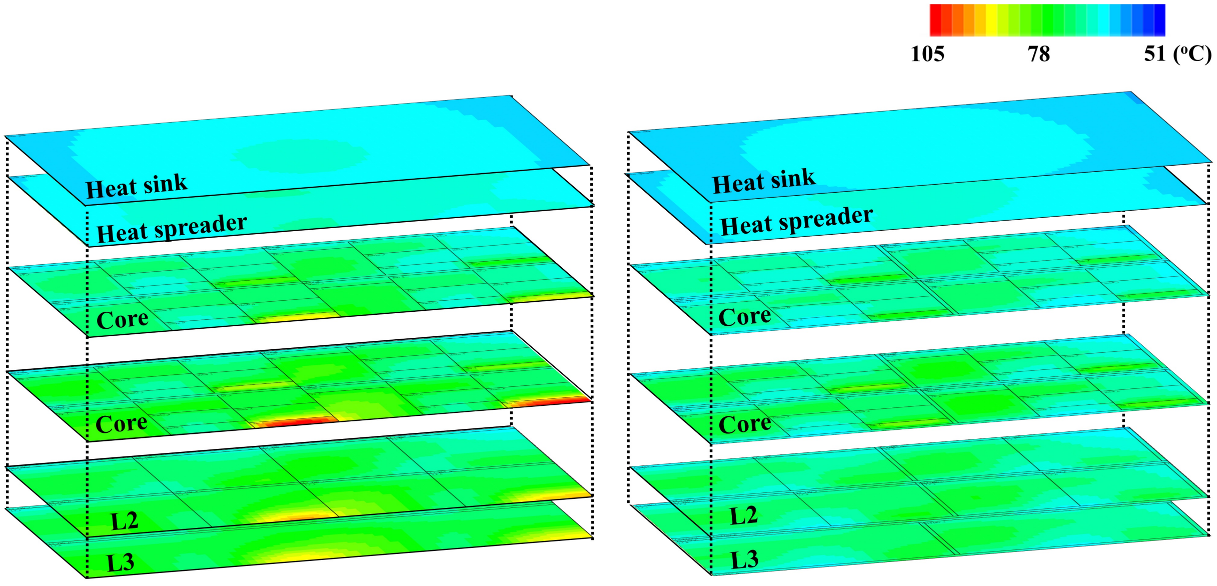


Figure 2.11: Maximum chip temperature of 2D Nehalem, 3D Nehalem and 3D Nehalem with TTSVs.



(a) 3D Nehalem without TTSVs

(b) 3D Nehalem with TTSVs

Figure 2.12: Temperature profiles of 3D Nehalem without/with TTSVs running BARNES benchmark. The BEOL layer and D2D layer are hidden for clarity.

Impact of Chip Frequency on 3D Nehalem temperature

To analyze the frequency impact on the thermal issues, we run the Splash-2 benchmarks with 1 GHz and 3 GHz on the 2D and 3D Nehalem and summarize the maximum chip temperatures in Figure 2.13. Results show that the peak temperature increases with frequency, especially for 3D Nehalem, which proves that the temperature increase is more significant with multi-core-layer 3D ICs due to the thermal bottleneck caused by the stacked hot spots and high thermal resistance interlayer materials. The placement and area overhead of TTSVs are the same as in Section 2.6.3, which are designed for 2 GHz chip frequency, and can substantially reduce the hot spot temperature for 3 GHz Nehalem but not below the threshold temperature (85 °C) for some benchmarks. It is expected that a larger TTSV area is needed to reduce the temperature further for higher frequencies.

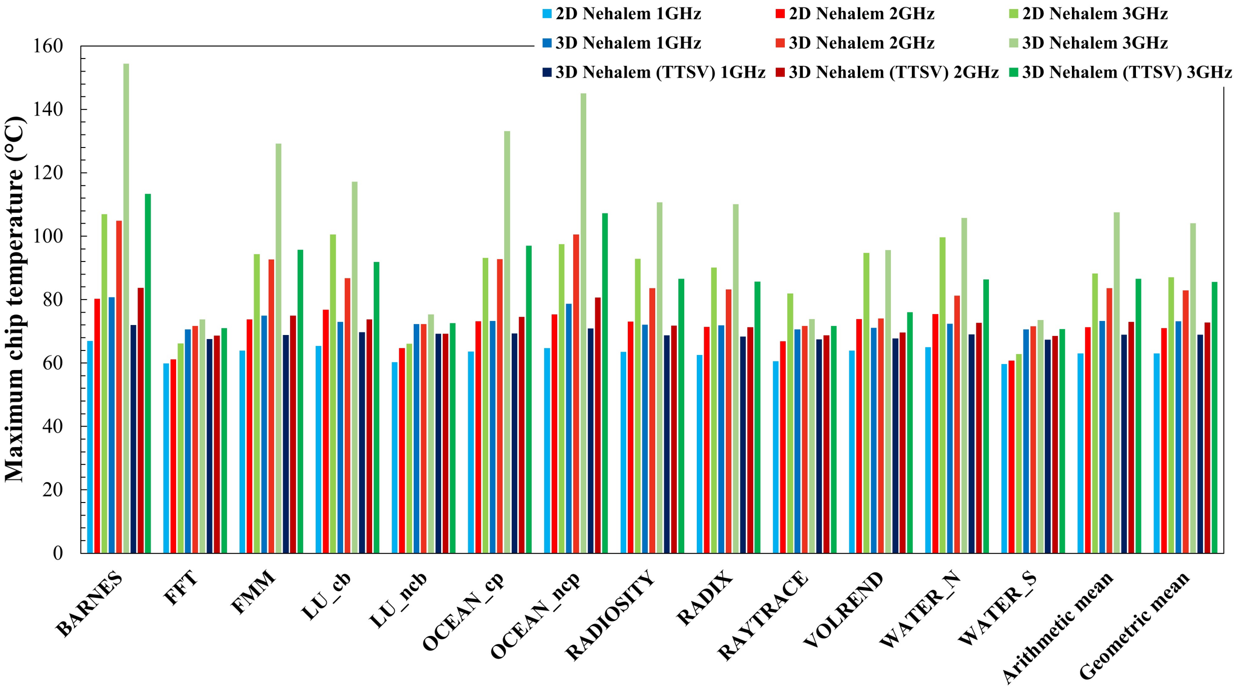


Figure 2.13: The maximum chip temperature of 2D Nehalem, 3D Nehalem and 3D Nehalem with TTSVs running Splash-2 benchmarks with 1 GHz, 2 GHz, and 3 GHz frequencies.

2.7 Conclusion

This chapter demonstrates a simulation flow that combines the system-level simulation and thermal simulation tools to provide an accurate thermal analysis of 3D ICs. We also propose a TTSVs placement method to assign the TTSV blocks along the lateral or vertical directions of the core layers and close to the hot spot regions, where TTSVs can further penetrate the cache layers to facilitate the heat dissipation. The simulation flow and TTSVs placement method are performed on the x86 based Nehalem floorplan; the results show that 3D Nehalem has considerable advantages in footprint, performance and power consumption. After adding the TTSVs blocks, the peak chip temperature is reduced by 5%-25% with an area overhead of 6%. Higher frequency also increases the hot spot temperature significantly for 3D ICs, a larger TTSVs area is expected for high-frequency applications.

Chapter 3

Thermal TSV Optimization and Hierarchical Floorplanning for 3D Integrated Circuits

While 3D ICs offer many advantages over 2D ICs, thermal management challenges remain unresolved. TTSVs are TSVs that facilitate heat transfer across stacked dies without carrying signal and provide a potential solution to thermal management in 3D ICs. However, the use of TTSVs can increase the distance between IC blocks in the floorplan and increase signal delay. The trade-off between temperature and wirelength is difficult to avoid in TTSV-integrated 3D ICs. Here we present a hierarchical approach to optimize the floorplan of a 3D Nehalem-based multicore processor via *Simulated Annealing* (SA) with respect to the area, temperature, and wirelength. Our simulations show that an increase in the TTSV area generally accompanies a decrease in the peak temperature, but the wirelength depends on the TTSV arrangement, which is uniquely optimized for each case of the allowed TTSV area. Compared to a simple method that places TTSVs between cores, our algorithm optimizes the TTSV arrangement between IC blocks and provides up to 17°C more reductions in peak temperature with the same TTSV area overhead of 20%. Compared to SA-optimized floorplans with no TTSVs, the TTSV-integrated SA-optimized floorplans offer up to 20°C more reductions in peak temperature with the TTSV area overhead of 20%, and at the same time, the wirelength increase is kept to 25%. The presented hierarchical floorplanning can effectively handle multiple constraints for advanced 3D multicore processors and provide optimal thermal management solutions.

This chapter is organized as follows: Section 3.1 gives an overview introducing the idea. The system-level simulation, power consumption calculation, and the architecture of the 3D Nehalem are presented in Section 3.2. The thermal model of the 3D Nehalem and the EMT-based analytical models of TSVs are provided in Section 3.3. The TTSV placement and the hierarchical floorplan algorithms are discussed in Section 3.4. Simulation results of temperature, wirelength, and area are analyzed in Section 3.5 and the whole chapter is concluded in Section 3.6.

3.1 Introduction

As the CMOS technology continuously scales down into the deep submicron regime and approaches the physical limits of minimization, the performance improvement through device scaling becomes more challenging [129]. 3D integration technology [50, 24, 138, 44] is based on interlayer connections and TSVs, which offers several benefits over 2D integration including higher packing density, reduced wire delay, less power consumption and heterogeneous integration [33, 110]. 3D ICs encounter inevitable thermal management problems [72, 128, 109, 16]. There are two main reasons for the thermal issues: 1) high power density due to a large number of active layers per unit area; 2) high thermal resistance between active dies and heat sinks in a stacked structure. Moreover, the aggressive wafer thinning process makes the thermal management of 3D ICs more challenging [112]. There have been various approaches to address the thermal issues in 3D ICs [74, 108]. Forced liquid cooling solutions based on single- or two-phase flow with micro-channels are effective in reducing the on-chip temperature [42, 122, 120, 76, 140], but the liquid cooling approaches may entail packaging issues. *ThermoElectric Cooling* (TEC) has received considerable attention as a solid-state cooling solution for local hot spots [15, 113]. However, TEC consumes extra power and the energy efficiency needs to be improved in order to be competitive [141]. On the other hand, thermal TSVs (TTSVs) are widely used to provide solid-state, passive, and local cooling. While signal TSVs (STSVs) are TSVs that carry both electrical signal and heat between dies [86, 4, 9], Thermal TSVs (TTSVs) are TSVs that do not carry signal but only facilitate heat transfer in 3D ICs. Applying TTSVs uniformly across the floorplan or specifically near hot spots can significantly reduce the peak temperature [86, 63, 4, 9]. However, the use of TTSV occupies additional routing space and increases the distance between IC blocks, which affects performance by increasing the power consumption and signal delay (e.g., $Delay = \frac{1}{2}R_{wire}C_{wire}WL$, where: R_{wire}, C_{wire}, and WL are the electrical resistance, capacitance and length of metal wires, respectively [6]). The corresponding trade-offs between temperature and performance need to be addressed for the TTSV placement. Wong and Lim [133] presented a SA-based floorplanning technique with a random walk based TTSV insertion algorithm, which achieves 17% temperature reduction with 3% TTSV density. However, inserting TTSV to the hottest units may not be possible because the TSVs are usually ten times large than logic gates and cannot be placed anywhere but only in the white space between IC blocks [87]. Li et al. [89] developed a two-stage TTSV insertion process: the vertical via distribution is solved by an analytical solution and the lateral via distribution is determined with SA-based floorplanning and white space redistribution. Zhao et al. [142] adopted integer multicommodity min-cost network to minimize the wirelength, TTSVs are also considered to reduce the temperature of superheated regions. The previous studies examined the TSV placement optimization with industrial circuit benchmarks such as ckt5 and ISCAS89 but neglect the lateral thermal properties of TSVs [133, 87, 89, 142, 106, 30, 67].

In our study, multithreaded application benchmarks are applied to the system-level simulator and the TTSV arrangement optimization occurs early in the design phase. To be more accurate with thermal modeling, we apply the *Effective Medium Theory* (EMT) to capture direction-dependent effective thermal conductivities of TSV units. This work presents a hierarchical floorplanning method with a TTSV arrangement optimization algorithm for 3D multicore processors. The contributions of this work are summarized as follows:

1. We conduct full system-level simulations using gem5 [22] to extract the component activities of Splash-2 benchmarks [135, 134] and calculate the power density of each IC block of a 3D Nehalem-based multicore processor (referred to as 3D Nehalem in the rest of the chapter) [84] using McPAT (Multicore, Power, Area, and Timing) [88]. We demonstrate that a critical benchmark (BARNES) can be selected to guide the floorplan optimization process.

2. We develop analytical models based on the EMT to capture lateral and vertical effective thermal conductivities of TSV unit cell. The impact of the liner layer is also considered and the results are validated by the *Finite-Element Method* (FEM).

3. We provide a TTSV placement algorithm based on SA, which optimizes the temperature, wirelength, and area of the floorplan. TTSV blocks with the direction-dependent thermal conductivities are considered during the optimization process.

4. We present a hierarchical approach to generate the floorplan of multicore 3D ICs, which utilizes an optimized single-core to complete the core layer floorplan of the multicore 3D IC based on a symmetric operation. The impacts of TTSV dimensions and area overheads on the peak temperature and wirelength of 3D ICs are discussed in detail.

3.2 System-Level Simulation and Power Consumption of the 3D Nehalem

For an accurate thermal analysis of a 3D multicore processor, the information about the power density for each IC block is required. In this section, we conduct system-level simulations to get the component activities using gem5 and calculate the power consumption of each IC block using McPAT.

3.2.1 System-Level Simulation Using gem5

Gem5 is adopted for the component activities, which is an event-driven system simulator with a modular platform for system-level architecture as well as processor microarchitecture research [22]. Splash-2 benchmarks are cross-compiled for the x86 (Nehalem) ISA [134]. The full system-level simulation is conducted for each benchmark to extract all the defined component activities, including the number of instructions for both integer and floating-point units, L1/L2/memory reads and writes with hit and miss rates, Network-on-Chip statistics and other related parameters.

3.2.2 Power Consumption Estimation Using McPAT

IC blocks data activity along with the system configurations are applied to McPAT, which is an integrated power, area and timing modeling framework for multithreaded, many core and multicore architectures [88]. The McPAT calculates dynamic power, static leakage power, and short circuit power.

3.2.3 The Architecture of the 3D Nehalem

We modified an x86 ISA out-of-order Nehalem processor, which is a family of *Intel Architecture* (IA) multicore processors based on a 45 *nm* technology [84]. Each core has one L1 instruction cache (32 KB), one L1 data cache (32 KB), and one private L2 cache (256 MB). A 16 MB L3 cache is shared between all the cores. The clock frequency is set to 2 GHz, which equals to a 0.5 ns periodicity. The architecture parameters are summarized in Table 3.1.

Table 3.1: x86 Nehalem-based processor parameter

Item	Description
Processor	x86 Nehalem-based 2 GHz processor with 8 cores
L1 icache	Private, 32 KB, 2-way set associate, 64 B blocks, 4 cycles latency, pseudo *Least Recently Used* (LRU) replacement.
L1 dcache	Private, 32 KB, 2-way set associate, 64B blocks, 4 cycles latency, pseudo LRU replacement.
L2 cache	Private, 256 KB, 4-way set associatice, 64B blocks, 12 cycles latency, with pseudo LRU replacement
L3 cache	Shared, 16 MB, 8-way set associative, 64B blocks, 30 cycles latency, with pseudo LRU replacement, MOESI cache coherence
Main memory	1 GB DRAM
Technology	45 *nm*

3.2.4 The Floorplan of 3D Nehalem

The 3D Nehalem is composed of two core layers with 4 cores in each layer, one L2 cache layer and one L3 cache layer are used to share data between threads. Figure 3.1a shows all the IC blocks in one core, which are consist of:

1. FP_0: includes the floating-point register file, floating-point units, and complex ALUs, which is responsible for divisions and multiplications.

2. Int_0: includes the integer register file, integer units, and integer ALUs, which is re-

sponsible for integer and logic operations.

3. I_cache_0 and D_cache_0: represent the instruction cache and the load/store data cache.

4. Itlb_0 and Dtlb_0: represent the instruction and data translation lookaside buffers.

5. Ifetch_0: is responsible for fetching instructions.

6. Others_0: includes the remaining blocks such as renaming units and ROB. The floorplan of the L2 cache is shown in Figure 3.1b, which contains all L2 caches for eight cores. The floorplan of the L3 cache is shown in Figure 3.1c. The core and cache layers are connected through Signal TSVs for electrical connection. For the 3D Nehalem with TTSVs, the area of L2 and L3 cache layers will be increased correspondingly for geometry matching.

3.3 Thermal Characterization and Modeling of 3d Nehalem and Tsvs

In this section, we demonstrate the thermal modeling of the 3D Nehalem and the analytical models used for effective thermal conductivities of TSV unit cells.

3.3.1 Thermal Modeling of the 3D Nehalem

The 3D Nehalem is constructed with the processor-on-top organization [4]. The two core layers are placed close to the heat sink for cooling purpose and the third and fourth layers are L2 and L3 caches as shown in Figure 3.2. STSVs are placed in four layers for signal communication and TTSVs are placed only in core layers for heat dissipation. The temperature profiles of 3D Nehalem with STSVs and TTSVs are simulated using HotSpot [70], which

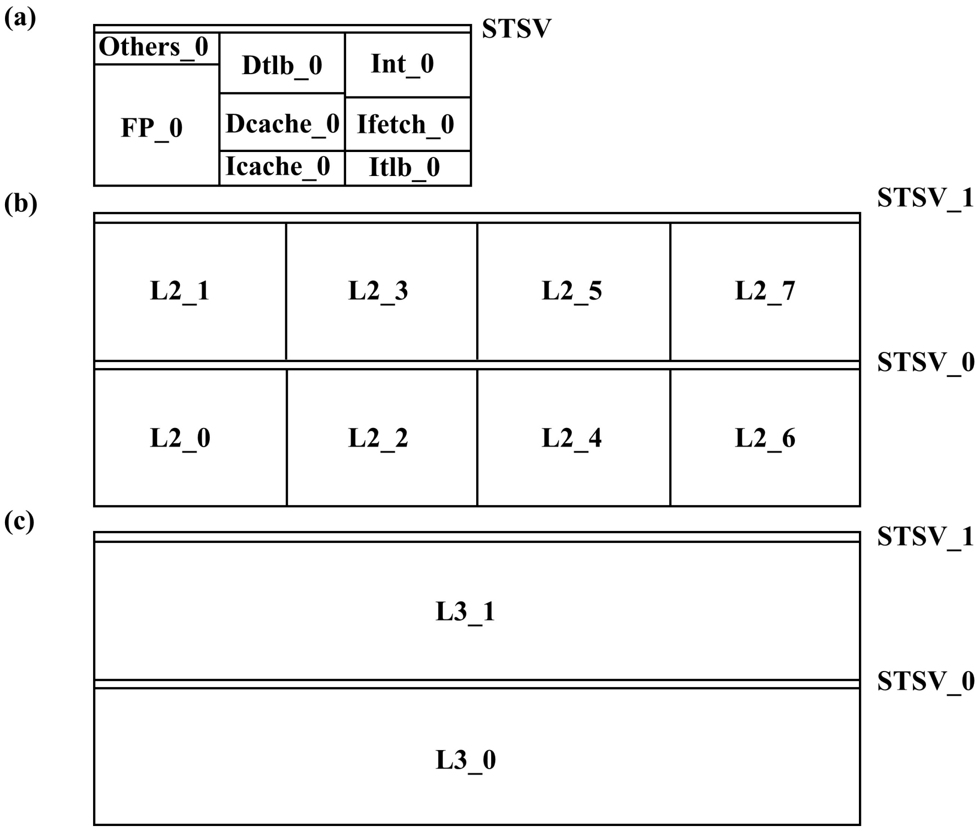


Figure 3.1: (a) The IC blocks and STSVs of $core0$ out of 8 cores for the 3D Nehalem. (b) The floorplan of the L2 cache. (c) The floorplan of the L3 cache. For 3D Nehalem with TTSVs, the area of L2 and L3 cache layers is increased for geometry matching.

utilizes a circuit-solving technique to solve an RC network of thermal resistances and capacitances by employing the thermal-electrical duality. Each layer is divided into grid cells, which is modeled with its own power density and thermal property as shown in Figure 3.2. HotSpot applies an air-cooling boundary condition on the heat sink with a convection resistance of 0.1 KW-1 and an ambient temperature of 45°C. The thicknesses of the silicon die and interconnect layers are assumed to be 100 μm and 20 μm, respectively [4, 9]. The TSV core is Cu because of its high electrical and thermal conductivities. The liner material of TSV is Si_3N_4, which is used for the isolation purpose with a thickness of 100 nm [119]. The materials properties used in thermal simulations are obtained from previous studies and are summarized in Table 3.2 [4, 49, 24, 91].

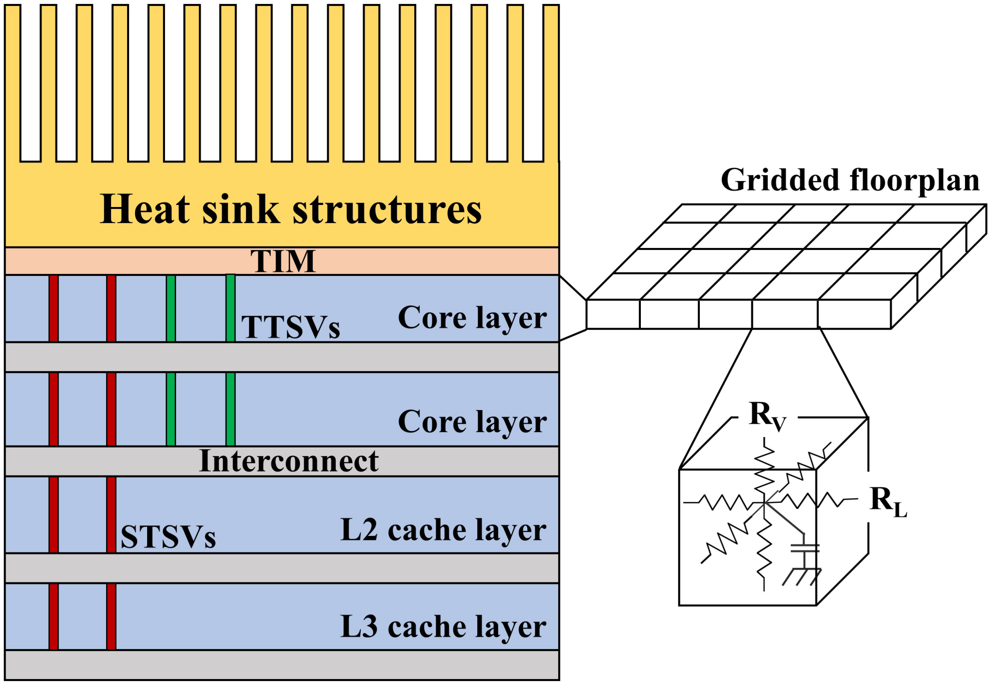


Figure 3.2: Package model of 3D Nehalem with two core layers, one L2 cache layer, and one L3 cache layer. The green block represents the TTSV, which are only presented in core layers for heat dissipation. The red block represents the STSV, which are placed in four layers for signal connection. The grid model is also shown, where each grid cell is modeled with one thermal capacitor and 6 thermal resistors with 4 resistors in the lateral direction and 2 resistors in the vertical direction.

Table 3.2: Material properties

Material	Thermal Conductivity ($Wm^{-1}K^{-1}$)
Silicon	100
Thermal interface material	5
Interconnect layer	1.5
Copper (TSV core)	400
Si_3N_4 (liner material)	30

3.3.2 Analytical Models of TSV Unit Cell

The schematic of the TSV farm is shown in Figure 3.3a, whose thermal conductivity is obtained using the TSV unit cell shown in Figure 3.3b. The pitch size (p) of the TSV farm is fixed to be 20 μm. For STSVs, the diameter (d) is assumed to be 10 μm as suggested in previous papers [143, 40]. With a fixed pitch size, the diameter of STSVs is mainly limited by the thermal stress induced carrier mobility change of the silicon near STSVs [73]. Since TTSV is inserted in white space, the diameter of TTSV can be larger. In our simulations, we use TTSV diameters of 10 μm and 15 μm to evaluate the cooling performance improvement with respect to larger TTSVs diameter. In the lateral direction, the thermal conductivity of the TSV unit cell is captured with a modified EMT model, where the impact of the liner layer is simplified as the thermal boundary resistance between Cu and Si. Assuming the heat flow is vertical to the liner layer as shown by the top-down view of the TSV unit cell in Figure 3.3c. The thermal resistance of the liner layer can be represented as [90]:

$$R_{liner} = \int_{r}^{r+\delta} \frac{dx}{2\pi x Len k_{liner}} \tag{3.1}$$

where r is the radius of the Cu core, δ is the liner layer thickness, x is the position variable across the liner layer and Len is the height of the TSV unit cell. The effective thermal conductivity of the TSV unit cell can be represented by the EMT as [65]:

$$k_{xTSVunit} = k_{Si} \frac{(\frac{k_{Cu}}{k_{Si}} - \frac{k_{Cu}}{R_c^{-1} r} - 1)\phi + 1 + \frac{k_{Cu}}{R_c^{-1} r} + \frac{k_{Cu}}{k_{Si}}}{-(\frac{k_{Cu}}{k_{Si}} - \frac{k_{Cu}}{R_c^{-1} r} - 1)\phi + 1 + \frac{k_{Cu}}{R_c^{-1} r} + \frac{k_{Cu}}{k_{Si}}} \tag{3.2}$$

where k_x TSV unit is the lateral thermal conductivity of TTSV unit cell. k_{Si} and k_{Cu} are the thermal conductivities of the Si substrate and Cu core. ϕ is the volume fraction of Cu core. R_c is the thermal boundary resistance, which is equivalent to $2\pi r L R_{liner}$. In the vertical direction, the thermal conductivity of the TSV unit cell is calculated as:

$$k_{zTSVunit} = k_{Si}(1 - \phi - \varphi) + k_{Cu}\varphi + k_{Si_3N_4}\phi \tag{3.3}$$

where φ is the volume fraction of the liner layer. The lateral and vertical thermal conductivities of TSV unit cell with varying TSV diameters are shown in Figure 3.3d. The analytical solutions fit well with the FEM simulations. The lateral thermal conductivity is much smaller compared to that of the vertical direction. Neglecting the lateral thermal property of TSV may be inaccurate in evaluating its cooling performance in 3D ICs.

3.4 TTSV Placement Optimization and Hierarchical Floorplanning Method

In this section, we explain the optimization flow and demonstrate algorithms for multipurpose floorplan optimization and hierarchical floorplanning.

3.4.1 Optimization Flow

The objective of this floorplan optimization method is to maximize the heat conduction of TTSV while simultaneously minimizing their negative impacts on wirelength and area. The whole optimization flow is shown in Figure 3.4.

The simulation flow is performed to optimize the floorplan of a single core and generate the corresponding 3D ICs with a fixed area of IC blocks and TTSV area overhead i. The number

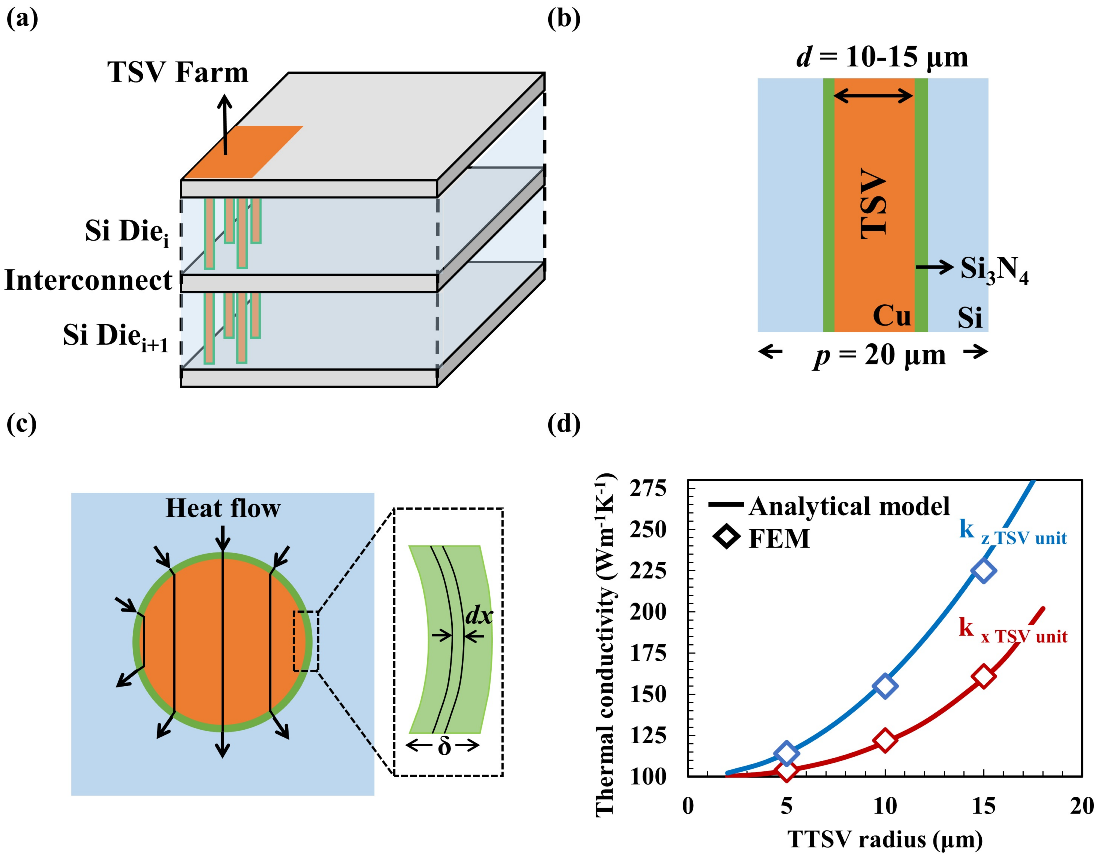


Figure 3.3: (a) Schematic of the TSV farm structure. (b) Schematic of the cross-sectional view of the TSV unit cell. (c) Schematic of the top-down view of the TSV unit cell, the black lines represent the heat flow, which is vertical to the liner layer and is straight inside the Cu core due to its high thermal conductivity. (d) The effective lateral and vertical thermal conductivities of TSV unit cell with different dimensions. The lines represent the analytical solutions and diamond squares represent the FEM results. Blue color represents the vertical direction and red color represents the lateral direction.

of TTSV blocks in one core is n and its maximum value is N as a bound of the simulation. If N is small, each TTSV block will be large which restricts its placement. If N is too large, the search space and computation time of SA will be extended significantly. We assume N = 16 for our simulation efforts.

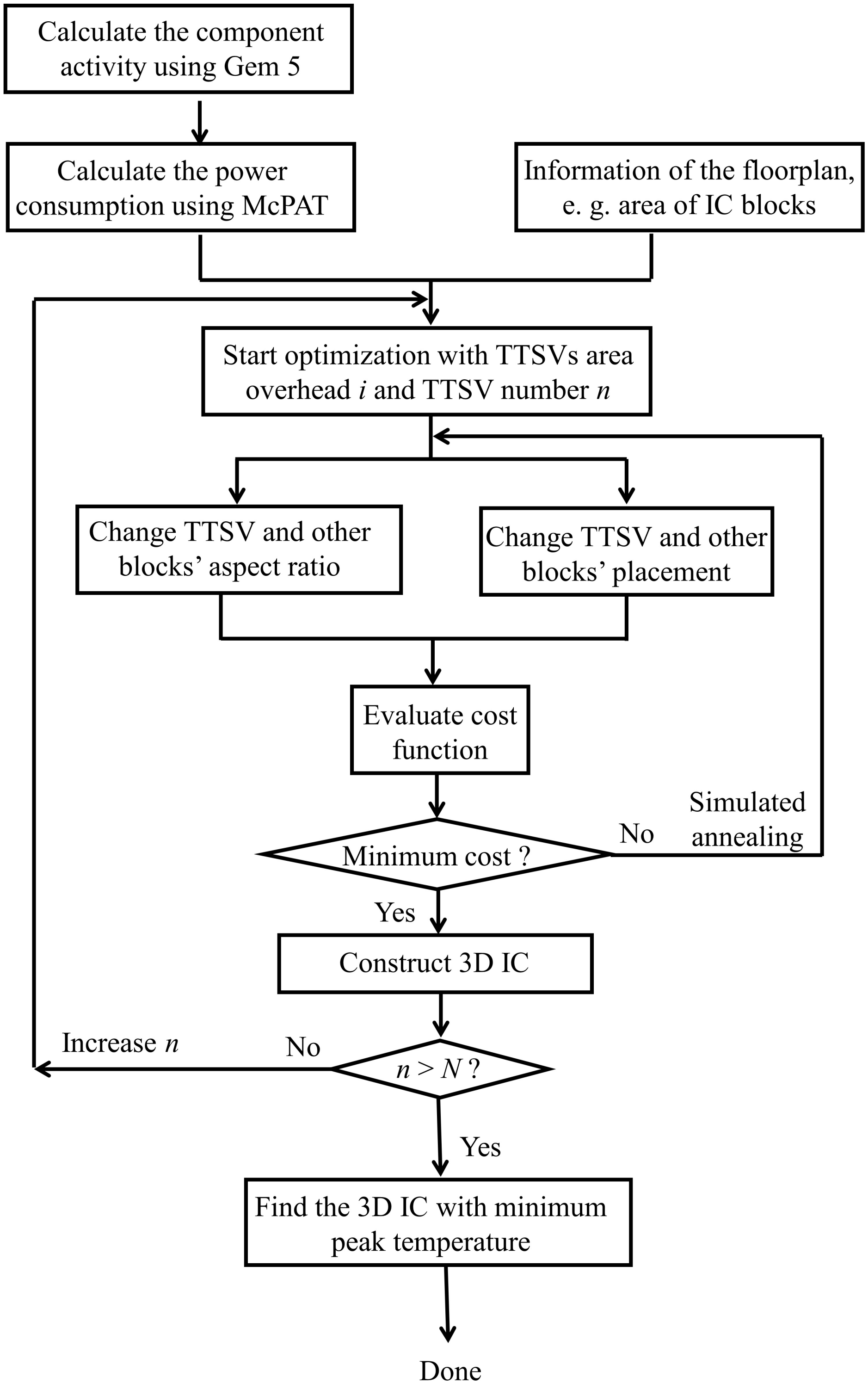

Calculate the component activity using Gem 5
Calculate the power consumption using McPAT
Information of the floorplan, e. g. area of IC blocks
Start optimization with TTSVs area overhead *i* and TTSV number *n*
Change TTSV and other blocks' aspect ratio
Change TTSV and other blocks' placement
Evaluate cost function
Minimum cost ?
No
Simulated annealing
Yes
Construct 3D IC
Increase *n*
No
n > *N* ?
Yes
Find the 3D IC with minimum peak temperature
Done

Figure 3.4: The floorplan optimization flow for 3D Nehalem

3.4.2 Problem Formulation for Floorplan Optimization

The goal is to find an efficient solution for TTSV placement early in the design process and minimize the total area, peak temperature and wirelength. Our assumptions for this work are: A *floorplan* (flp) consisting of:

- m blocks $B_{0:m-1}$, namely, $B_0, B_1, \ldots, B_{m-1}$. Each block has its associated area $A_{0:m-1}$, namely, $A_0, A_1, \ldots, A_{m-1}$ with a restricted acceptable minimum and maximum aspect ratio.

- n TTSV blocks $TTSV_{0:n-1},$, namely, $TTSV_0, TTSV_1, \ldots, TTSV_{n-1}$. Similarly, each block has its associated area, $TTSVA_{0:n-1}$, namely, $TTSVA_0, TTSVA_1, \ldots, TTSVA_{n-1}$ with restricted acceptable minimum and maximum aspect ratios.

- Vector A, representing the area of all blocks, is the concatenation of $A_{0:m-1}$ and $TTSVA_{0:n-1}$

- Wire density matrix WD of size $m \times m$, $WD = \begin{pmatrix} 0 & w_{01} & \ldots & w_{0m} \\ w_{10} & 0 & \ldots & w_{1m} \\ \ldots & w_{ij} & w_{ii} & \ldots \\ w_{m0} & w_{m1} & \ldots & w_{mm} \end{pmatrix}$, and Manhattan distance matrix L of size $m \times m$, $L = \begin{pmatrix} 0 & l_{01} & \ldots & l_{0n} \\ l_{10} & 0 & \ldots & l_{1n} \\ \ldots & \ldots & l_{ij} & \ldots \\ l_{n0} & l_{n1} & \ldots & 0 \end{pmatrix}$ between blocksB_i and $B_j \in B_{0:m-1} \forall i, j \in [0, m-1]$.

Our algorithm calculates the following:

- IC block activities, which represent the number of instructions executed in all functional blocks, the number of all reads and writes in cache units and are calculated using gem5.

- Power consumption for each blocks $B_0, B_1, \ldots, B_m$, which is estimated and stored in P containing $P_0, P_1, \ldots, P_m$, using McPAT.
- The peak temperature of each block, which is generated using HotSpot and stored in T represented as: $T_0, T_1, \ldots, T_{n+m}$.

The objective function is represented by:

$$Optimal\_flp\,(B_{0:m-1}\ ,\ TTSV_{0:n-1}\ , A,\ WL,\ P) \tag{3.4}$$

subjects to a cost function:

$$F_{cost} = \alpha A + \beta WL +\ \gamma T \tag{3.5}$$

where α, β and γ are the weight factors with the units of m^{-2}, m^{-1}, and °C^{-1}, respectively. A is the total area of the floorplan and T is the peak temperature, which is used to reduce the white space and evaluate the cooling performance, respectively. WL is added to limit the negative impact of TTSV on the performance, which is defined as:

$$WL = \frac{1}{2}\sum_{i=0}^{m-1}\sum_{j=0}^{m-1} w_{ij} l_{ij} \tag{3.6}$$

The corresponding 3D structure is created using $Create\_3D\_structure(flp\_best)$, where flp_{best} is the best floorplan generated by $Optimal\_flp()$ with all IC blocks represented by their width, height, and position in the floorplan as a (x, y)pair.

3.4.3 Simulated Annealing Based Optimization Algorithm

The SA-based floorplan optimization algorithm is *Optimal_flp*, which is used to optimize the block placement for a single core. The output of optimized floorplan is fed to the hierarchical floorplanning algorithm, Create_3D_structure, to construct the 3D ICs. Algorithm 1 (*Optimal_flp*), accepts the floorplan description including processor parameters (see Table 3.1) as an input, and generates the optimized floorplan based on SA, flp_{best}. Line 1 denotes all the gem5 generated activities in set $B_{0:m-1}$, which are fed to McPAT to estimate the static and dynamic power consumption (stored in P) on Line 2. Lines 3-29 comprehensively describe the SA process with parameter values and expressions listed in Table 3.3. On Line 3, we populate an initial floorplan in flp_c using normalized polish expression and calculate its F_{cost} shown on Line 4. Lines 5-6 set flp_{best}, $cost_{best}$ and the annealing temperature, T, to their initial values. The annealing process stopping condition is reaching either the maximum number of allowable steps, 3000, or the cooling limit restricted by T_{cold} as shown in line 7. Lines 8-11 present that the number of block movements initialized on Line 8, governs the generation of the random floorplan, flp_{new}, on Line 10.

The new floorplan, flp_{new}, is produced for testing, if the calculated cost, $cost_{new}$, on Line 11 is moving downhill in the optimization curve or has been tried twice the total amount of the block movements. If the new generated floorplan cost, $cost_{new}$, is less than the current one, then the cost function downhill movement section, which is represented by Lines 12 -16, stores the best floorplan so far in flp_{best} with its associated $cost_{best}$. Otherwise, current floorplan is given a chance with acceptance likelihood of the Boltzmann probability function$e^{-\frac{cost_{new}-cost_c}{T}}$ in the uphill movement from Lines 17-21 or it will be rejected on Line 22. If at any time the ratio of rejection, $Rreject$, is not satisfied, the annealing process for the current iteration is

Table 3.3: Simulated annealing parameter

symbol	Description	Value/expression
$Steps_{max}$	Maximum number of iterations	3000
P_0	Initial Probability	0.99
D_{avg}	Average change in the cost function	1
$Rreject$	Rejection ratio to stop annealing	0.99
$Rcool$	Ratio of annealing cooling schedule	0.99557
flp_c, flp_{best}	Current floorplan, Best floorplan	
flp_{new}	New floorplan	
$cost_c$, $cost_{best}$	Current cost, Best cost	
$cost_{new}$	New cost	
Mv	No. of block moves per step	15
Mvs	All blocks movements	$Mv\ .\ m$
$T_{initial}$	Initial Annealing Temperature	$-\frac{D_{avg}}{log(P_0)}$
T	Current Annealing Temperature	
T_{cold}	Cooling Temperature	$-\frac{D_{avg}}{log(\frac{1-Rreject}{2})}$
ΔF_{Cost}	$F_{cost\_new} - F_{cost\_old}$	

terminated on Line 25. The best floorplan stored at flp_{best} is returned on Line 29.

3.4.4 Hierarchical Floorplanning of the 3D Nehalem

In this section, the pseudo code of the Algorithm 2 ($Create\_3D\_structure$) for the hierarchical floorplanning is explained. The final output is the constructed multicore 3D IC with four cores in each core layer, whose temperature distribution can be represented by a single core because of the thermal symmetry. The Algorithm 2 receives the output of the Algorithm 1, flp_{best}, as $core0$ and composed of four parts. The first part identifies the position of the peak temperature region (hot spot) within $core0$ on Line 1, which will be in one of the four possible positions shown in Figure 3.5a. The second part transforms the $core0$ orientation to position 1 to separate hot spots, which are the black dots in Figure 3.5a. To change $core0$ orientation, let (x_{c0}, y_{c0}) denote the left bottom edge of $core0$. If it is already in position 1, no additional step is needed (Lines 2- 3). Lines 4-6, shown in Figure 3.5b, check if $core0$ is in position 2. If true, then $core0$ is flipped along the x axis by changing (x_{c0}, y_{c0}) to $(x_{c0}, -y_{c0})$,

Algorithm 1 *Optimal_flp*: SA-based floorplan optimization algorithm

```
Require: $flp$ description explained in problem formulation and processor parameter (e.g. Table 3.1)
Ensure: the best optimized floorplan using SA, $flp_{best}$
 1: Activities←Generate activities using gem5 with processor parameter in Table 3.1
 2: $P \leftarrow$ Generate power traces using McPAT
            //initializations
 3: $flp_c \leftarrow Initialize\_flp()$
 4: $cost_c \leftarrow Fcost(flp_c, P, \alpha, \beta, \gamma)$ //Eq. 3.5
 5: $flp_{best} \leftarrow flp_c$ ; $cost_{best} \leftarrow cost_c$
 6: $T \leftarrow T_{initial}$
            //SA
 7: for $s \leftarrow 0$ ; $T >= Tcold$ and $s < Steps_{max}$ ; $s++$ do
 8:     $Mvs \leftarrow Mv(m+n)$ ; $Downs \leftarrow 0$ ; $Rejects \leftarrow 0$
 9:     for $q \leftarrow 0$ ; $q < 2Mvs$ and $Downs < Mvs$ ; $q++$ do
10:         $flp_{new} \leftarrow make\_random\_move(flp_c)$
11:         $cost_{new} \leftarrow Fcost(flp_{new}, P, \alpha, \beta, \gamma)$
            // SA downhill movement
12:         if $cost_{new} < cost_c$ then $Downs++$
13:             if $cost_{new} < cost_{best}$ then
14:                 $flp_{best} \leftarrow flp_c$ ; $cost_{best} \leftarrow cost_{new}$
15:             end if
16:             $flp_c \leftarrow flp_{new}$ ; $cost_c \leftarrow cost_{new}$
            // SA uphill movement
17:         else if $rand() < e^{-\frac{cost_{new}-cost_c}{T}}$ then
18:             if $cost_{new} < cost_{best}$ then
19:                 $flp_{best} \leftarrow flp_c$ ; $cost_{best} \leftarrow cost_{new}$
20:             end if
21:             $flp_c \leftarrow flp_{new}$ ; $cost_c \leftarrow cost_{new}$
22:         else$Rejects++$
23:         end if
24:     end for
25:     if $Rejects/Tries > Rreject$ then break
26:     end if
27:     $T \leftarrow T * Rcool$
28: end for
29: Return $flp_{best}$
```

and shifted up by adding the $core0$ height (h) to the y axis as $(x_{c0}, -y_{c0} + h)$. Lines 7-9 are shown in Figure 3.5c. If $core0$ is in position 3, it is flipped along the y axis by changing (x_{c0}, y_{c0}) to $(-x_{c0}, y_{c0})$, then shifted right by adding the $core0$ width (w) to the x axis as $(x_{c0} + w, y_{c0})$. For Lines 10-14, if $core0$ is in position 4, the combination of positions 2 and 3 actions is performed (Figure 3.5d) by flipping along the x axis and shifting up, then flipping along the y axis and shifting right, and (x_{c0}, y_{c0}) becomes $(-x_{c0} + w, -y_{c0} + h)$. The third part constructs the 2D layer as shown in Figure 3.5e and Lines 15-20, by duplicating *core*0, flipping it along the x axis and adding its height twice, (x_{c1}, y_{c1}) is changed to $(x_{c1}, -y_{c1}+2h)$. After generating (*core*0, *core*1) pair, the pair is duplicated, flipped along the y axis, and then shifted right twice to generate the layer with 4 cores (*core*0, *core*1, *core*2, and *core*3). In the fourth and last part (Line 21), the 3D structure is generated by copying and stacking (*core*0, *core*1, *core*2, and *core*3) to (*core*4, *core*5, *core*6, and *core*7) (see Figure 3.5f).

Algorithm 2 *Create_3D_structure*: hierarchical floorplanning for the 3D Nehalem.

```
Require: the best optimized floorplan using SA, $flp_{best}$ (i.e. core0 in Figure 3.5a)
Ensure: 3D structure of four cores in one layer and another four cores in another layer.
 1: position ← find the core quarter containing the maximum temperature block. //Figure 3.5a
    //position the maximum temperature blocks in the bottom left corner on core0
 2: if position = 1 then
 3:     //do nothing
 4: else if position = 2 then          //Figure 3.5b i
 5:     flip core0 along x axis         //Figure 3.5b ii
 6:     shift core0 up by its height    //Figure 3.5b iii
 7: else if position = 3 then          //Figure 3.5c i
 8:     flip core0 along y axis         //Figure 3.5c ii
 9:     shift core0 right by its width  //Figure 3.5c iii
10: else//position = 4                  //Figure 3.5d i
11:     flip core0 along x axis         //Figure 3.5d ii
12:     shift core0 up by its height    //Figure 3.5d iii
13:     flip core0 along y axis         //Figure 3.5d v
14:     shift core0 up by its width     //Figure 3.5d vi
15: end if
    //construct a 2D layer with 4 cores
16: copy core0 to a new core1
17: flip core1 along x axis             //Figure 3.5e ii
18: shift core1 up by its height ×2
19: copy (core0, core1) to a new (core2, core3)
20: flip (core2, core3) along y axis    //Figure 3.5e iv
21: shift (core2, core3) up by its width ×2
    //construct a 2D layer with 4 cores
22: 3D_structure ← copy (core0, core1, core2, core3) to a new layer (core4, core5, core6,core7)
23: Return 3D_structure
```

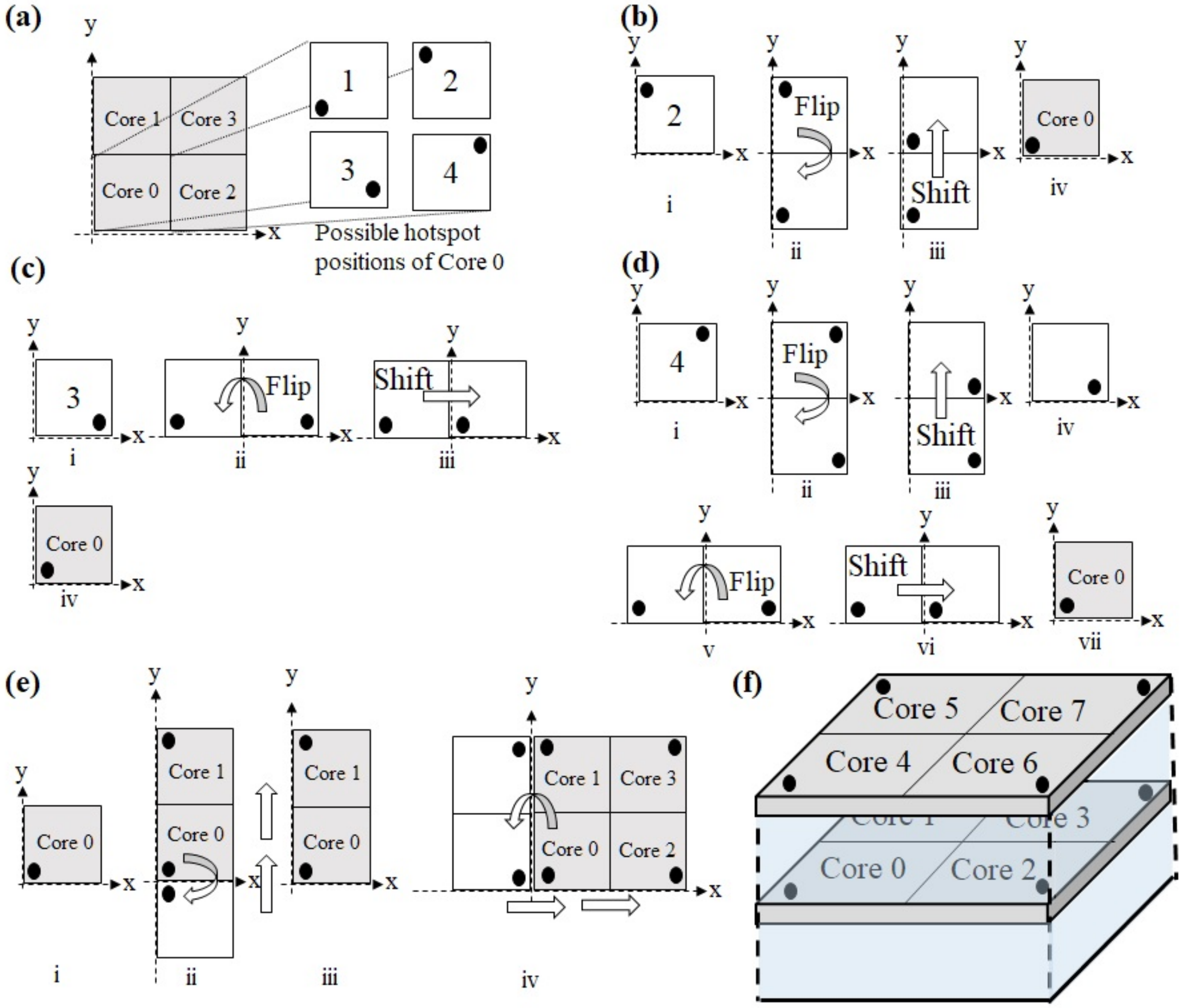


Figure 3.5: (a) Four cores in the x, y coordinates with all possible maximum temperature block $B_{T_{max}}$ positions of $core0$ (b) Operations required to transform $B_{T_{max}}$ from position 2 to 1 (c) Operations required to transform $B_{T_{max}}$ from position 3 to 1 (d) Operations required to transform $B_{T_{max}}$ from position 4 to 1 (e) Operations required to create 2D with four cores: $core0$, 1, 2 and 3 (f) the resulted 3D structure (only in-core layers are shown).

3.5 Results and Discussion

In this section, the power densities of the 3D Nehalem running Splash-2 benchmarks are presented and the effectiveness of our floorplan optimization method is evaluated. Also, the impacts of TTSV on the peak temperature, wirelength, and area of multicore 3D ICs are studied.

3.5.1 Power Density of the 3D Nehalem Running Splash-2 Benchmarks

The power consumptions of Splash-2 benchmarks are extracted from McPAT based on the component activities obtained from gem5 [22]. Figure 3.6 shows the power density of each IC block in one core. Among all the benchmarks, the maximum power density occurs running BARNES, which even exceeds 500 Wcm^{-2}. Among all the IC blocks, *itlb* has the maximum power density running most of the benchmarks, which makes it the critical hot spot in the 3D ICs. TTSVs are placed close to *itlb* during the optimization process to facilitate heat removal. The optimal floorplan with TTSV will provide effective passive cooling to most of the benchmarks as well. In the following simulations, BARNES is used to guide the floorplan optimization with different TTSV dimensions and area overhead. The optimized floorplans are adopted to calculate the peak temperatures of all Splash-2 benchmarks to evaluate their effectiveness.

3.5.2 Impacts of TTSVs on the Core Layer Peak Temperature of the 3D Nehalem

To evaluate the performance of our floorplan optimization method, the peak temperatures of TTSV-integrated 3D Nehalem with/without TTSVs placement optimization are compared. First, the SA and hierarchical floorplanning are performed on the 3D Nehalem with no TTSV. Then, TTSVs are added between cores without further optimization as the non-optimized fixed TTSV arrangement. We use 15 μm diameter TTSV for the non-optimized fixed TTSV arrangement case in thermal simulations as the best-case scenario. Figure 3.7a shows floorplan of one core at 2% TTSV area overhead. For larger area overheads, the width of the TTSV blocks will increase without changing the arrangement. For comparison, the SA-based algorithm and hierarchical floorplanning are performed considering TTSV blocks.

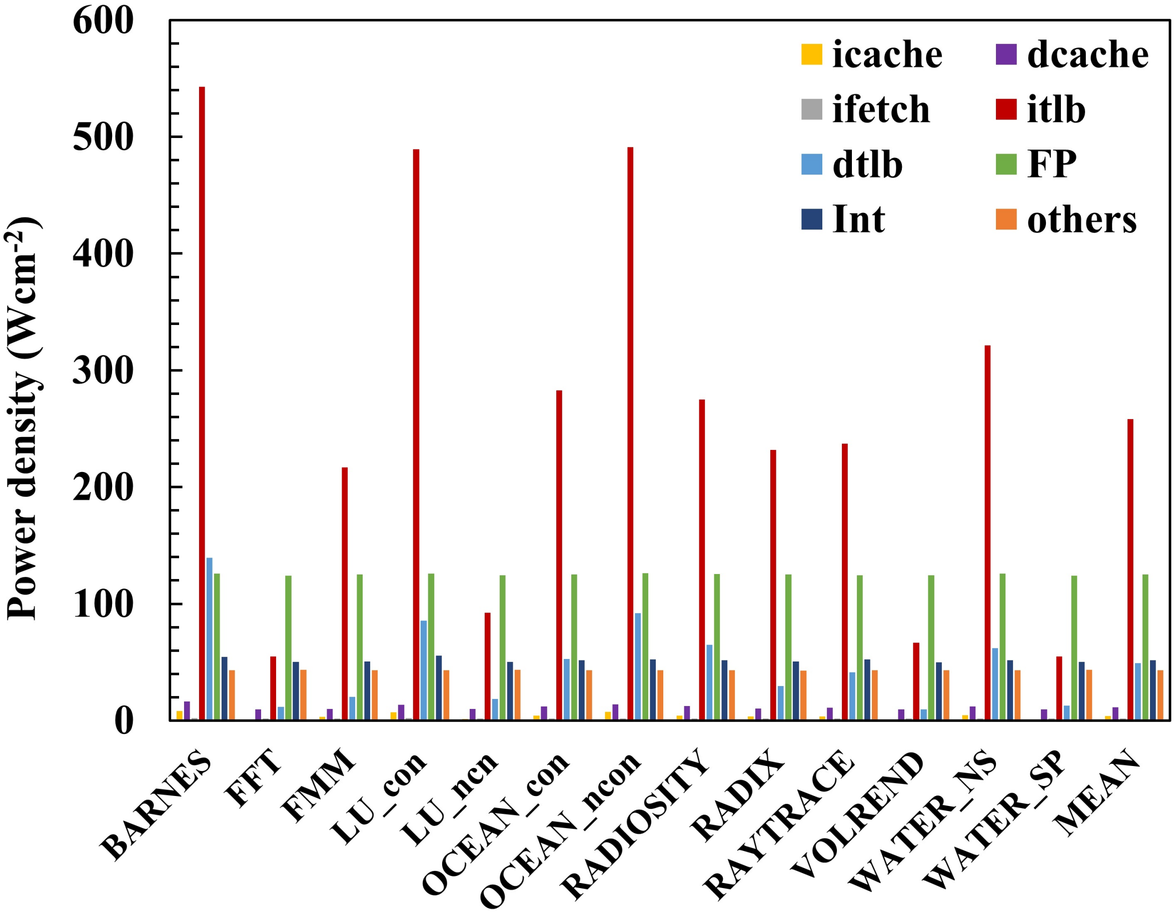


Figure 3.6: The power densities for Splash-2 benchmarks for all the IC blocks in one core of the 3D Nehalem.

Figure 3.7b shows the optimized floorplan of one core with 10 μm diameter TTSVs at 2% area overhead. TTSVs located between the IC blocks and around *itlb* (hot spot) to facilitate heat transfer. The optimized floorplan of one core with 15 μm diameter TTSVs at 2% area overhead is shown in Figure 3.7c.), where the TTSVs are also placed around *itlb* block to facilitate heat dissipation.

As shown in Figure 3.2, the second core layer attributes the maximum on-chip temperature because of its high-power density and a large thermal resistance to the heat sink. The corresponding second core layer temperature profiles of the 3D Nehalem running BARNES benchmark are compared in Figures 3.7d, 3.7e and 3.7f. With the same TTSV area overhead, the peak temperatures of the 3D Nehalem with TTSV placement optimization are 7°C and 10°C lower compared to that of the non-optimized fixed TTSV arrangement case when TTSV diameters are 10 μm and 15 μm, respectively. Apart from the *itlb*, the temperatures of other IC blocks are close, which means that the optimal arranged TTSVs provide localized cooling to hot spots. The temperature profiles of the entire 3D Nehalem are shown in Figures 3.7g, 3.7h, and 3.7i with non-optimized TTSV arrangement, 10 μm diameter, and 15 μm diameter optimized TTSV arrangement, respectively. By optimizing the TTSV placement inside the core layers, the peak temperatures of the core layers are reduced as well as the temperatures of L2 and L3 cache layers, which means that placing TTSVs only in the core layer can provide effective cooling to the entire 3D ICs including the cache layers.

Figure 3.8 shows the peak temperatures of the non- optimized TTSV arrangement case and optimized TTSV arrangement cases at different TTSV area overheads running BARNES benchmark. With the TTSV placement optimization, peak temperatures of optimized floorplans are much lower than that of the non-optimized TTSV arrangement case with 10 μm and 15 μm diameter TTSVs. With the same pitch size, the temperature reduction is more substantial for 15 μm diameter TTSV because of the high effective thermal conductivity and the average peak temperature reduction is 5°C lower than that of the 10 μm diameter TTSV. The peak temperature is constantly reduced with increasing area overhead. When the TTSV area overhead is at 2%, the peak temperature reductions are 7°C, 10°C for 10 μm, and 15 μm diameter TTSVs while the peak temperature reduction is 0.3°C for the non-optimized fixed TTSV arrangement case. When the TTSV area overhead is at 20%, the peak temperature reductions are 16°C, and 20°C for 10 μm and 15 μm diameter TTSVs while the peak temperature reduction is 3.5°C for the inter-core TTSV arrangement case. Our floorplan optimization method is effective in reducing the peak temperature in multicore 3D ICs.

Using BARNES benchmark to guide the floorplan optimization, the peak temperature is

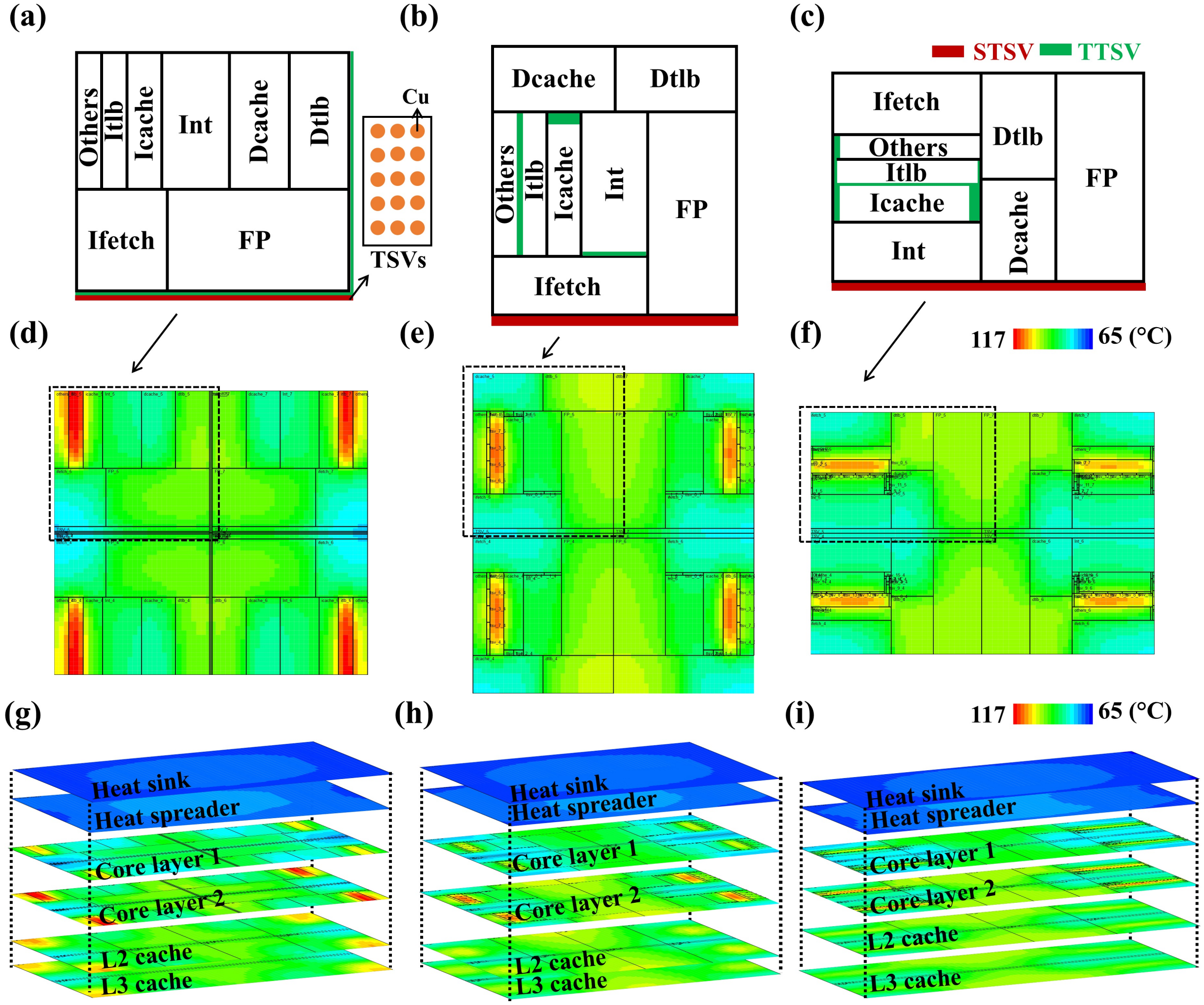


Figure 3.7: (a), (b), and (c) Single-core floorplans of the 3D Nehalem with non-optimized or fixed TTSV arrangement and optimized TTSV arrangement. The TTSV diameters are 15 μm , 10 μm , and 15 μm , respectively. (d), (e) and (f) are the temperature profiles of the second core layer of the 3D Nehalem for the non-optimized or fixed TTSV arrangement case, 10 μm-, and 15 μm-diameter TTSVs optimized cases running BARNES benchmarks. The corresponding placement of (a), (b), and (c) are shown by black dashed squares. (g), (h), and (i) are the corresponding temperature profiles of the entire 3D Nehalem, the interconnect layers are hidden for clarity.

much lower than the non-optimized TTSV arrangement case. To evaluate the effectiveness of the optimized floorplans, we further conduct thermal simulations of the 3D Nehalem running other Splash-2 benchmarks. We use floorplans at 2% TTSV area overheads (the floorplans are shown in Figure 3.7 because they provide the maximum temperature reductions per area overhead, which are 3.5°C and 5°C per 1% TTSV area overhead for 10 μm

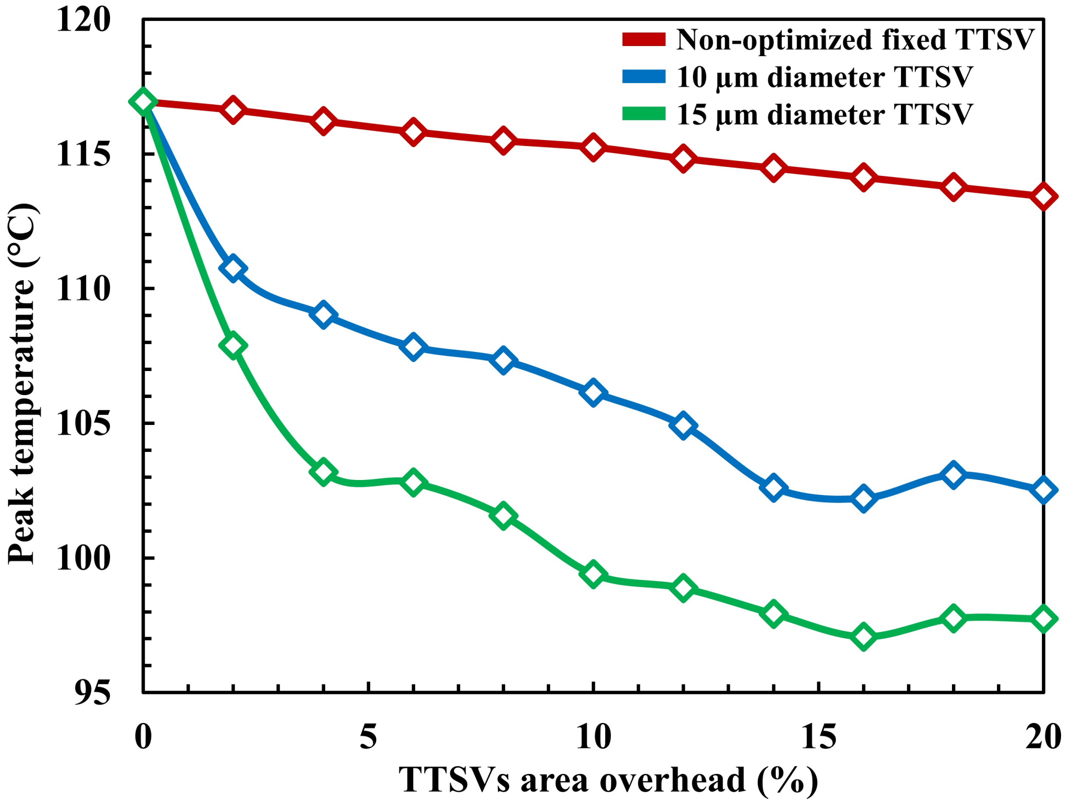


Figure 3.8: Peak temperature change for the 3D Nehalem with/without the TTSV placement optimization running BARNES benchmark. The peak temperature of the 3D Nehalem is generally reduced with the increase of TTSV area overhead. The cooling performance of 15 μm diameter TTSV is much better compared to that of the 10 μm diameter TTSV. The area overhead of TTSV is varied from 0% to 20%.

and 15 μm diameter TTSVs. The results are shown in Figure 3.9. The peak temperatures of Splash-2 benchmarks that are larger than 100°C are reduced by 5 to 10°C for the two different TTSV dimensions, which indicate that the optimized floorplans based on BARNES are effective for other Spahs-2 benchmarks as well. Some benchmarks have increased peak temperature because of the changed floorplan (e.g., high-power-density IC blocks of this benchmark are put closer), but those benchmarks' peak temperatures are much lower and the peak temperature increase is negligible (less than 3°C) for both 10 μm and 15 μm diameter TTSVs. The detail information about maximum temperature, minimum temperature, and average temperature of the second core layer are summarized in Table 3.4. The average temperatures are close for the three cases while the peak temperatures of the optimized floorplan are much lower, which proves the effectiveness of our method for local hot spot cooling in 3D ICs.

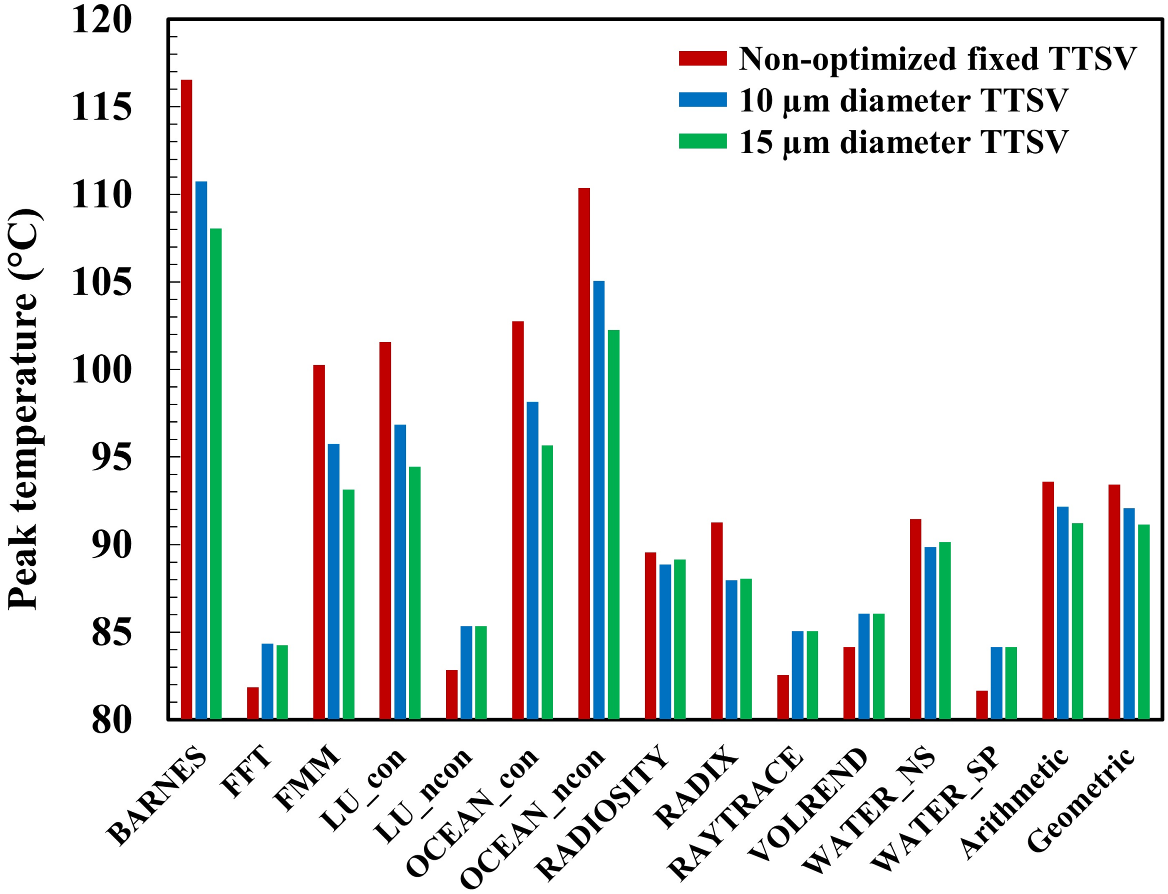


Figure 3.9: Peak temperatures for the 3D Nehalem with/without the TTSVs placement optimization running Splash-2 benchmark. The non-optimized fixed TTSV arrangement and optimized TTSV arrangement are considered with TTSV diameters of 15 μm, 10 μm, and 15 μm, respectively. The floorplans are the same as shown in Figure 3.7. The arithmetic mean and the geometric mean are also shown in the figure.

3.5.3 Impacts of TTSVs on Area and Wirelength of 3D Nehalem

The total area increases linearly with the TTSV area overhead because of the negligible white space. The change of wirelength is strongly dependent on the TTSV arrangement, which is uniquely optimized for each TTSV area overhead. The optimized floorplan without TTSV

Table 3.4: The maximum, minimum, and average temperature of splash-2 benchmarks for the second layer of 3D Nehalem in (℃)

Benchmarks	Non-optimized fixed TTSVs			10 μm-diameter TTSV			15 μm-diameter TTSV		
	max T (℃)	min T (℃)	avg T (℃)	max T (℃)	min T (℃)	avg T (℃)	max T (℃)	min T (℃)	avg T (℃)
BARNES	116.6	74.5	90	110.8	76.8	90.2	108.1	77.2	89.7
FFT	81.9	61.1	69.4	84.4	59.9	69.6	84.3	60.7	69.3
FMM	100.3	66	76.3	95.8	64.9	76.5	93.2	66	76.1
LU_con	101.6	69	80.7	96.9	69	80.9	94.5	69.9	80.4
LU_ncon	82.9	62.1	70.7	85.4	60.6	70.9	85.4	61.4	70.6
OCEAN_con	102.8	69	80.6	98.2	68.4	80.8	95.7	69.3	80.4
OCEAN_ncon	110.4	72.1	85.7	105.1	73.6	85.8	102.3	74.3	85.4
RADIOSITY	89.6	66.7	77.3	88.9	65.6	77.4	89.2	66.4	77.1
RADIX	91.3	65.3	75.2	88	63.7	75.4	88.1	64.7	75
RAYTRACE	82.6	59.3	70	85.1	59.5	70.2	85.1	60.2	69.9
VOLREND	84.2	61.9	71.7	86.1	61.1	71.9	86.1	62	71.6
WATER_NS	91.5	67.7	78.3	89.9	67	78.5	90.2	67.8	78.1
WATER_SP	81.7	61.1	69.2	84.2	59.9	69.4	84.2	60.8	69.1
Arithmetic	93.6	65.8	76.5	92.2	65.3	76.7	91.2	66.2	76.3
Geometric	93.4	65.8	76.4	92.1	65.3	76.6	91.1	66.1	76.3

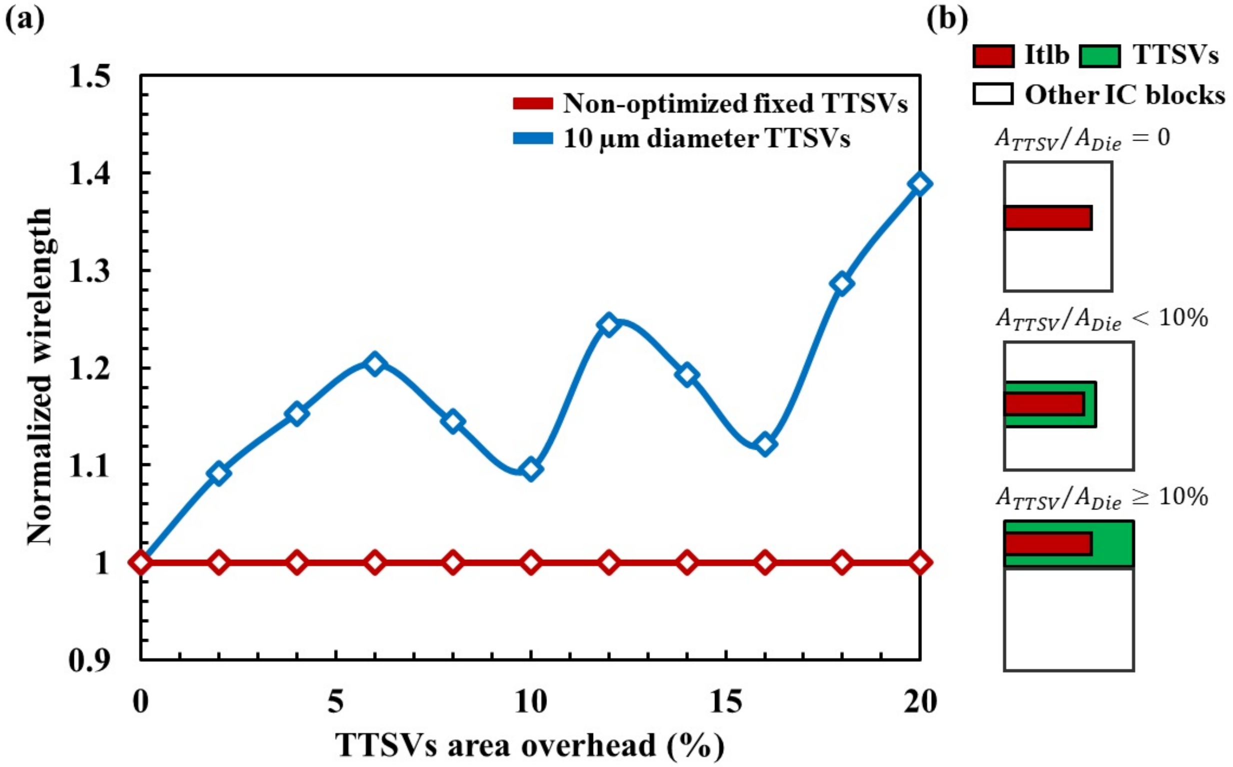


Figure 3.10: (a) The normalized wirelength change with different TTSV area overheads, the TTSV diameters are 10 μm and 15 μm, respectively. The wirelength of the no-TTSV case or inter-core TTSV placement case is considered as 1 to normalize the wirelength. (b) The simple schematics of the floorplan of one core with different TTSV area overheads. The red blocks indicate the *itlb*, the green blocks indicate the TTSV and the white blocks indicate other blocks inside one core.

and with non-optimized TTSV arrangement both have the minimum wirelength because there is no TTSV between the IC blocks inside the core. Their wirelength is considered as 1 to normalize the wirelength of the TTSV optimization case. Figure 3.10a shows the change of the wirelength with 10 μm and 15 μm diameters TTSVs at different area overheads. A simple schematic of the optimized floorplan at 0% TTSV area overhead is shown in Figure 3.10b (top part), where the hot spot, *itlb*, is surrounded by other IC blocks inside the core. As the TTSV area overhead increase (less than 10%), TTSVs are placed around the *itlb* for heat dissipation, which increases the wirelength by enlarging the distance between IC blocks.

The schematic of optimized floorplan at small TTSV area overhead is also shown in Fig-

Table 3.5: Normalized wirelength and area

TTSV area overhead	10 μm TTSV		15 μm TTSV	
	Wirelength	Area	Wirelength	Area
0%	1.000	1.000	1.000	1.000
2%	1.091	1.02	1.109	1.039
4%	1.153	1.04	1.155	1.045
6%	1.205	1.065	1.123	1.058
8%	1.145	1.094	1.024	1.080
10%	1.096	1.113	1.248	1.128
12%	1.244	1.126	1.287	1.125
14%	1.193	1.168	1.164	1.151
16%	1.122	1.205	1.215	1.175
18%	1.286	1.21	1.147	1.224
20%	1.389	1.236	1.247	1.243

ure 3.10b (middle part). As the TTSV area overhead becomes larger (more than 10%), the *itlb* is moved to the edge of the core to reduce the negative impacts of TTSV on the wirelength. Part of the TTSV is located at the core edge to facilitate heat removal without increasing the wirelength. Because of the large TTSV area overhead, the wirelength will still increase induced by the TTSV part that still remains between the *itlb* and other IC blocks, which results in an increasing trend of the wirelength with respect to TTSV area overhead. Even though there is an increasing trend of the wirelength in Fig. 10(a), the fluctuation is attributed to the effectiveness of the optimization especially at 10% and 16%, which is related to the location of the hot spot and the TTSV area overhead. The schematic of the optimized floorplan at large TTSV area overhead is shown in Figure 3.10b (bottom part). The normalized wirelength and area of TTSV-integrated 3D Nehalem with 10 μm and 15 μm diameters at different TTSVs area overhead are listed in Table 3.5.

3.6 Conclusions

This chapter demonstrates a hierarchical floorplanning approach for a 3D Nehalem-based multicore processor, which optimizes the peak temperature, wirelength and area of the floorplan through an SA-based TTSV placement optimization algorithm. Our simulation results show that optimally arranged TTSVs can effectively reduce the peak temperature with moderate sacrifices in wirelength and area overhead. The peak temperature decreases consistently with the TTSV area overhead while the wirelength change is strongly related to the TTSV placement, which is uniquely optimized with different TTSV area overheads. Moreover, a critical benchmark can be selected for guiding the floorplan optimization, and the optimized floorplan is applicable to other Splash-2 benchmarks without further modification.

Chapter 4

Online Monitoring and Adaptive Routing for Aging Mitigation in NoCs

Scalability of NoC as a promising solution for many-core systems can be jeopardized because of reliability challenges such as aging in advanced silicon technology. Previous mitigation techniques to protect NoC are either offline, while aging is strictly influenced by runtime operating conditions, or impose significant overheads to the system. This work presents an online monitoring method through a *Centralized Aging Table* (CAT) for routers in NoCs. Router's capacity in flits, which are the main stimuli in routers, is predictable and limited for a given period of time. Consequently, stress rate and temperature, which are the major sources of aging mechanisms such as *Bias Temperature Instability* (BTI) and *Hot Carrier Injection* (HCI), will be in the predictable ranges, as well. Hence, our methodology uses CAT which is populated by values that represent aging degradation for each different pairs of stress and temperature ranges during a given period of time. Furthermore, utilizing CAT, we propose an online adaptive aging-aware routing algorithm in order to avoid highly aged routers which eventually leads to age balancing between routers. Additionally, our proposed routing algorithm reduces the maximum age of routers by changing the shortest paths between source-destination pairs adaptively, considering routers' ages across them in each given period of time. Extensive experimental analysis using gem5 simulator demonstrates that our online routing algorithm and monitoring methodology, CAT, improves delay degradation of maximum aged router and aging imbalance on average by 39% and 52% compared to XY routing, respectively. The impact of our proposed methodology on network latency, *Energy-Delay-Product* (EDP) and link utilization is negligible.

This chapter is organized as follows: an overview is introduced in Section 4.1. In Section 4.2, we elaborate and discuss related works. After that, an overview of the impact of aging mechanisms in NoC is demonstrated in Section 4.3. Our proposed aging monitoring technique using CAT is presented in Section 4.4. Section 4.5 proposes our aging-aware routing algorithm. Then, the experimental setup and results are discussed in Section 4.6. Finally, the chapter is concluded in Section 4.7.

4.1 Introduction

Delay degradation generated by aging mechanisms becomes a reliability challenge in advanced semiconductor technology [2]. It imposes a large design margin to the critical paths which results in design complexity and overhead [92, 75]. BTI and HCI are two dominant aging mechanisms causing accelerated transistor aging, which increase the transistor threshold voltage (V_{th}) over time [2, 92, 75]. This impacts the lifetime of the chip in the long term and its performance (or critical path) in the short term. Consequently, threatening the performance and scalability for many-core designs. Therefore, NoC that consist of packet-switched routers for providing high bandwidth, parallelism, and scalability for many-core systems requires careful aging investigation.

Critical path's age is affected by operating conditions such as stress (i.e. usage of transistors along it) and temperature, which change with time because of variation in running an application on a system. In other words, higher temperature and stress leads to higher aging rate. Since flits are the only router's stimuli, change in number of flits (fl) inside the router and their residence time (rs) affect the temperature and stress of the router. By monitoring fl and rs we can predict temperature and stress; thus aging rate. Additionally, fl and rs are stimulated by the routing algorithm. Hence, the router's age and reliability have direct relations with the routing algorithm. By changing the routing algorithm and controlling source-destination shortest paths selection, the aging impact on NoC can be mitigated.

Since router as a component in NoCs, has a predictable and limited capacity of flits for a given period of time (P), then stress (S) and temperature (T) as two main sources of BTI and HCI are in limited ranges. Considering this observation, we propose a methodology based on a Centralized Aging Table (CAT). CAT is populated by the amount of aging degradation for different ranges of fl and rs in a router from zero up to the router capacity. This makes CAT independent of the running application. CAT, which is stored in one of the cores, can be accessed by all routers in the NoC in order to accumulate their current age to the pre-evaluated respective aging degradation. To compute fl and rs, a counter and a timer is embedded into each router (elaborated in Section 4.4).

As shown in Figure 4.1, routers with different usage (i.e. different fl and rs) experience different temperatures and stresses, thus are impacted by aging differently. This leads to an imbalance in age of routers, which may lead to reliability and scalability challenges in NoC. In this chapter, we proposed an aging-aware routing algorithm which selects shortest paths between destination-source pairs adaptively based on the router's age using CAT. CAT helps our proposed algorithm to adapt the shortest paths online periodically as opposed to state-of-the-art works, which adapt routing based on offline aging information through profiling [19, 20]. Therefore, our routing algorithm finds k-best shortest paths and selects between them periodically based on the impact of aging on routers (using CAT) to reduce maximum aged router and balance the age between routers (elaborated in Section 4.5). Since aging mechanisms impacts critical path's delay gradually, we update the routing tables in periodic time (P) (e.g. each week). Our extensive experimental analysis using gem5 simulator reveals that the proposed aging-aware routing algorithm reduces maximum routers' age and imbalanced routers' age by 39% and 52% on average in different benchmarks, respectively. Since we select the best shortest path between k-best shortest paths (i.e. with same latency cost but different aging costs), the network latency overhead is negligible.

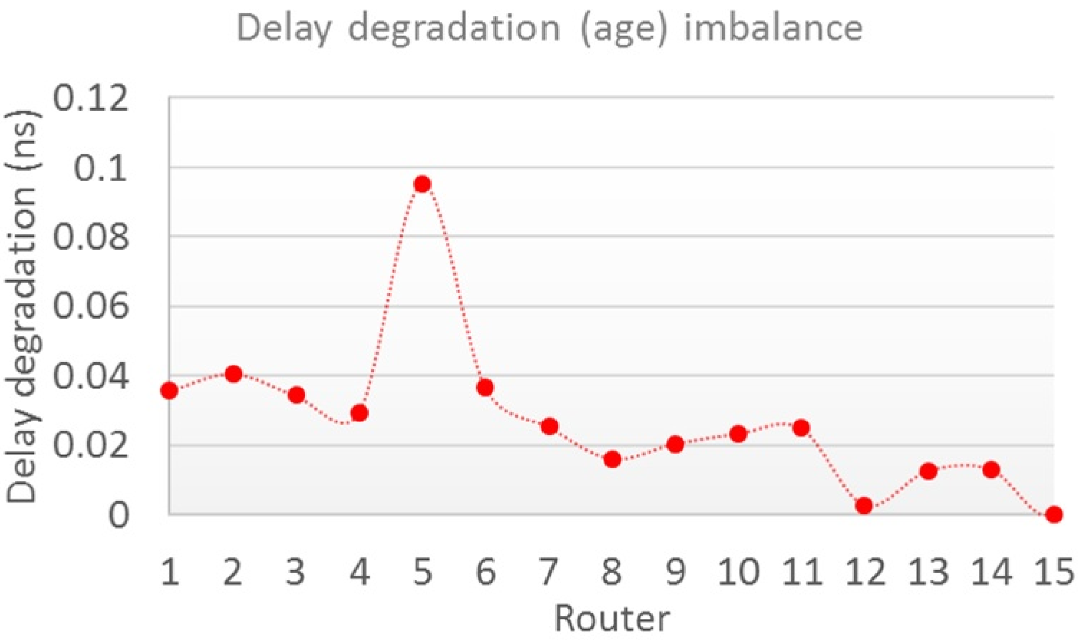


Figure 4.1: Age imbalance of different routers in FFT

4.2 Related Work

A multi-objective *Integer Linear Programming* (ILP) based routing algorithm is proposed in [19]. This technique assigns lifetime budgets to each router offline and using ILP finds the best route considering aging, power estimation, and performance. Since the lifetime budget assignment is offline, any change through online variation in workload cannot be captured. Authors in [20] proposed an adaptive aging aware algorithm considering the assigned lifetime budget to each router. Although their proposed routing algorithm adapts online, the lifetime budget that is assigned to each router is offline, which may overestimate or underestimate actual workload. The main shortcoming of these two solutions is that the lifetime budgets are assigned offline by profiling, and they are application dependent. Authors in [11] propose *Wearout Monitoring System* (WMS) for different components of a router to monitor aging online. Based on the packet's criticality, their algorithm chooses between buffered or bufferless routers to mitigate aging by deflecting non-critical packets to bufferless routers. This technique not only induces hardware overheads to routers due to complex WMSs, but also is only applicable in specific type of networks with different router architectures (i.e. heterogeneous NoC). Determining which packet is critical is also a crucial decision which may induce overhead to the system. A *Dynamic Programming* (DP) based routing algorithm is proposed in [131], which requires a parallel DP network as overhead to the system to propagate the lifetime budget of each router inside the network. Additionally, a complicated circuitry added to routers to find the lifetime budget of them which induce hardware overhead to the system, as well. The main shortcoming of these two solutions is large overheads. Authors in [98, 57] proposed aging aware task mapping for many-core heterogeneous architectures and a scalable sensor design that can be utilized in many core systems, respectively.

4.3 Aging in NoC

Transistors' delay degradation (Δd) (i.e. increment in V_{th}) manifests itself as delay degradation of critical paths along them. Thus, it results in timing failure or performance degradation of the system. In this section, we describe how induced Δd associated with BTI and HCI is computed for routers in NoC.

4.3.1 BTI Aging Effect

The available models for BTI [102, 132] describe it in two phases: stress phase and recovery phase. During the stress phase (i.e. transistor is ON and in high temperature), BTI occurs because of generation of the interface traps at Si-SiO_2, which gradually increase V_{th}. During the recovery phase (i.e. transistor is OFF) some of these traps are eliminated and partially recover the shift on V_{th}. Based on [102, 132, 71], the delay degradation caused by BTI can be simplified as:

$$\Delta d_{BTI} = C_{BTI} \times Y^n \times t^n \times e^{-(\frac{E_a}{kT})} \times d_0 \tag{4.1}$$

Where, d_0 is pre-aged delay of the transistor, t is the transistor age, Y is the duty cycle of the transistor (how long the transistor is ON), T is temperature, n is constant depending on the fabrication process, E_a is activation energy, k is Boltzmann's constant and C_{BTI} is BTI fitting parameter which also depends on the fabrication process.

4.3.2 HCI Aging Effect

By changing the current-voltage characteristic of transistor induced by accelerated carrier within electric field inside transistor channel, HCI increases the V_{th}. Based on [126, 124], the HCI delay degradation can be simplified as:

$$\Delta d_{HCI} = C_{HCI} \times \alpha \times f \times t^{0.5} \times e^{-(\frac{E_a}{kT})} \times d_0 \tag{4.2}$$

Where, α is the switching activity of the transistor, f is clock frequency, C_{HCI} is HCI fitting parameter which depends on fabrication process and the remaining symbols are as represented in Eq. 4.1. Duty cycle (Y) and activity factor ($\alpha \times f$) are both considered as stress (S) for BTI and HCI aging mechanisms, respectively.

4.4 Online Aging Monitoring in NoC

Based on Eq. 4.1 and Eq. 4.2, the delay degradation (Δd) of transistors generated from BTI and HCI are exponential function of temperature (T) and non-linear function of transistor usage, the so called Stress (S) [57, 102, 132, 71, 126]. Since the only stimuli in routers are flits, we leveraged the flits residence time (rs) inside the router and number of flits (fl) to predict and monitor router's age online. Stress (S) and average temperature (T) of a router are function of number of flits (fl) inside routers and their resident time (rs) (i.e. how long a router is busy) during a given period of time, epsilon (ϵ). Furthermore, considering the NoC characteristics (e.g. flit injection rate, topology, etc.), the maximum number of flits as well as their maximum residence time in is bounded by FL_{max} and RS_{max}, respectively. Therefore, we monitor fl and rs, which can be utilized to map their corresponding temperatures and stresses of a specific router.

For instance, as shown in Figure 4.2, each period P is divided to smaller period of ϵ. Therefore, each P is equal to ($n \times \epsilon$). For a specific ϵ (e.g. 10,000 cycles), fl is equal to 250 and rs is equal to 6,000 cycles ($RS_{max} = \epsilon =$ 10,000 cycles). This pair of fl and rs corresponds to a specific temperature, stress, and consequently aging rate. Figure 4.4, illustrates our proposed architecture to monitor fl and rs which is embedded into router architecture [1]. Each core i is connected to a router r_i. A parallel 12-bit counter [123] counts fl for each ϵ(upper counter in Figure 4.2). It monitors valid incoming flits to the router from different ports to the router using valid (V) and ready (R) signals. More will be elaborated in Section 4.6.

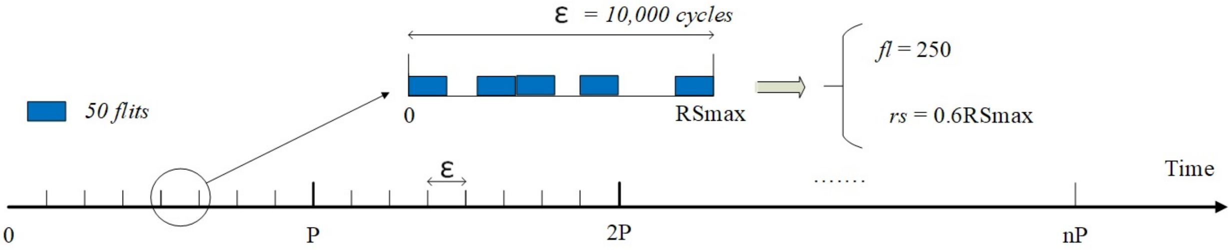
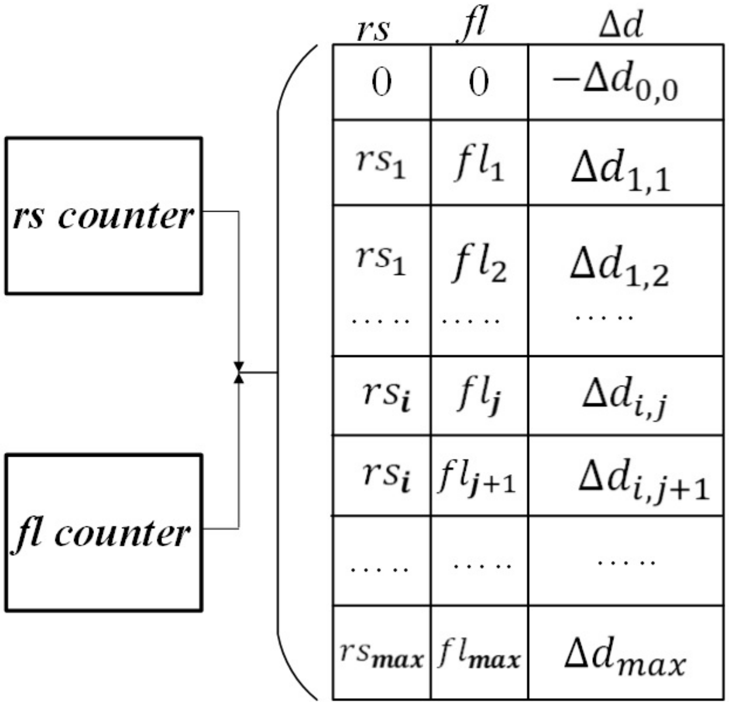


Figure 4.2: Aging monitoring for each period of P

The second parallel counter (lower counter in Figure 4.2) is a 14-bit parallel counter that counts the resident time (rs) of the flits. Since RS_{max} (i.e.ϵ) can be presented by 14 bits, the counter is preceded by a 14-bit subtractor to subtract the exit time of an outgoing flit, which is the current cycle when the flit is exiting the router, from the en-queue time, which is saved inside the flit when it is en-queued inside input buffer. we assume the maximum rs of a flit inside a 5-stage router is 15 cycles. Therefore, following the subtractor, a 4-bit MUX is connected to drop any possible negative subtractions in the boundaries of each 10,000 cycles. Then it is fed to the parallel counter to keep accumulating resident time (rs) of all flits exiting the router through all possible five output ports. Moreover, these two counters reset after ϵ cycles. We use a timer to count ϵ and whenever it reaches to ϵ a reset signal is sent to the two parallel counters inside each router to be ready for next ϵ.

To minimize the distance between CAT and all routers, CAT must be located in one of the middle routers. Figure 4.2) illustrates that CAT resides in core 5. CAT will be accessed using (fl, rs) pair from all routers to read back their age degradation in each ϵ. Age degradation of a router can be computed for each temperature and stress (Eq. 4.1 and Eq. 4.2). To this end, we determine conditions that may happen to a router. Each condition, $C_{i,j}$, is represented by its respective rs_i and fl_j. Each pair of (rs_i, fl_j) corresponds to temperature $T_{i,j}$ and stress $S_{i,j}$ (i.e. $(T_{i,j}, S_{i,j})$). Hence, each condition is a function of rs_i and fl_j and each condition corresponds to a specific aging rate. For example, in Figure 4.3, when number of flits is fl_2 and they reside inside the router for queueing, processing and traversal through



rs	fl	Δd
0	0	$-\Delta d_{0,0}$
rs_1	fl_1	$\Delta d_{1,1}$
rs_1	fl_2	$\Delta d_{1,2}$ .....
rs_i	fl_j	$\Delta d_{i,j}$
rs_i	fl_{j+1}	$\Delta d_{i,j+1}$
.....	.....	.....
rs_{max}	fl_{max}	Δd_{max}

Figure 4.3: CAT and required counters

the router for rs_1 cycles out of ϵ cycles, the delay degradation is $\Delta d_{1,2}$. If the router is not busy (fl and rs are equal to zero) BTI recovery phase happens and CAT is filled by a negative corresponding amount of recovery.

Algorithm 3 shows how CAT is constructed. The inputs to this algorithm are maximum residence time RS_{max} (or the updating time period (ϵ)), the steps for each residence time rs_{steps}, the number of steps for counting flits inside the router fl_{steps}, and the injection rate to the system, $Ijrate$. The algorithm's output is CAT which can be accessed from each router to read back its own age based on fl and rs during each ϵ. At the beginning, the maximum number of flits (FL_{max}) that can occupy a router during RS_{max} (or ϵ) considering maximum $Ijrate$ is extracted (line 1). The list of residence time (rs) and number of flits (fl) will be created based on their number of steps (line 2, 3). Using these two lists we calculate power consumption that can be used in HotSpot [70] for temperature extraction map (line 6). Each different residence time rs_i and number of flits fl_j have different power and temperature maps ($T_{i,j}$). Similarly, the stress will be extract as $S_{i,j}$ based on HCI and

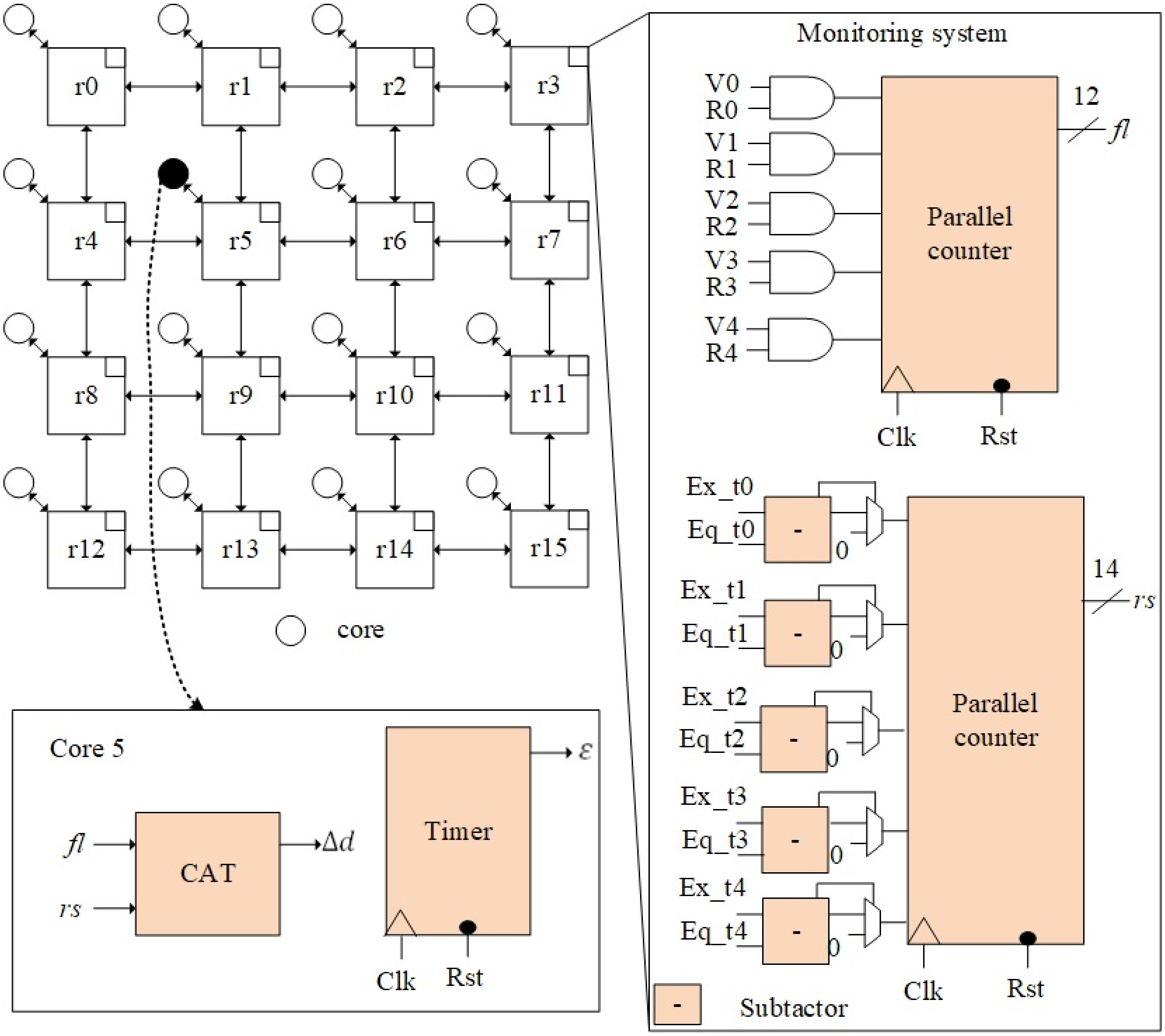


Figure 4.4: Proposed online monitoring

BTI aging mechanism (line 7). As shown in Eq. 4.1 and Eq. 4.2, Stress (S) is a function of duty cycle (Y) in BTI and switching activity (α) multiplied by clock frequency (f) in HCI. In this work, Eq. 4.3 is utilized to calculate S as follows:

$$S = m_1 \times Y + m_2 \times \alpha \times f \tag{4.3}$$

Since impact of BTI is higher than HCI, m_1 is three times greater than m_2. For BTI, Y is equal to rs and for HCI α is equal to the ratio between fl and FL_{max}. The delay degradation for each pair of temperature and stress are extracted as $\Delta d_{(i,j)}$ using Eq. 4.1 and Eq. 4.2 (line 8). Finally, the CAT will be updated of flits residence time (rs_i), number of flits (fl_j), and their corresponding delay degradation ($\Delta d_{(i,j)}$).

Algorithm 3 CAT construction

```
Require: Maximum resident time $RS_{max}$, number of resident time steps $rs_{steps}$, number of
    flits steps $fl_{steps}$, injection rate $Ijrate$.
Ensure: {CAT}
 1: $FL_{max} \leftarrow FindMaxFlit(Ijrate, RS_{max})$;
 2: $\{rs\} \leftarrow CreateRsList(RS_{max}, rs_{steps})$;
 3: $\{fl\} \leftarrow CreateFlList(FL_{max}, fl_{steps})$;
 4: for all $rs_i \in rs$ do
 5:     for all $fl_j \in fl$ do
 6:         $P^k_{(i,j)} \leftarrow CalPower(rs_i, fl_j)$;
 7:         $T^k_{(i,j)} \leftarrow CalTempreture(P^k_{(i,j)}, RFLP)$;
 8:         $S_{(i,j)} \leftarrow CalStress(rs_i, fl_j)$;
 9:         $\Delta d^k_{(i,j)} \leftarrow CalDelayDeg(T^k_{(i,j)}, S_{(i,j)})$;
10:         CAT$\leftarrow FillCAT(rs_i, fl_j, \Delta d_{(i,j)})$;
11:     end for
12: end for
13: Return {CAT};
```

4.5 Aging-Aware Routing Algorithm

As mentioned earlier, operating conditions and different fl and rs in routers during a specific time lead to aging rate difference in routers. This imbalanced aging may result in timing failure in the highly aged (i.e. used) router and impact the scalability and reliability of the system. There are different shortest paths between source-destination pairs or paths that are very close to the shortest paths in term of cost (delay). These paths use different routers to transfer flits inside the network. Algorithm 4 proposes an aging-aware routing algorithm using an added tag to the routers as their age. The age tag in each router will be updated online using CAT, periodically ($P = n \times \epsilon$). This tag is leveraged for choosing the best aging aware shortest path between all available shortest paths from each source-destination pairs. The routing table in each router will be updated adaptively at each period of time P. The inputs to Algorithm 4 are list of source-destination pairs, $(Src, Dest)$ and list of routers' age, RAg. The algorithm's output is the list of shortest paths for each source-destination pairs, $ShortPathPair$. Using $CalShortestPath()$, we find k-best shortest paths list for each pair of source-destination. Dijkstra's shortest path algorithm is leveraged to find this list. There are different algorithms that can be utilized for this purpose [51, 8]. After that, for each pair we check which paths do not include the maximum aged router by calling$MaxAgeR()$ and then find the best paths based on the minimum summation of ages on their routers using the list of ages by calling $MinAge()$ (line 7). The new shortest paths for each source-destination pairs are found for the nextϵ and are added to the list of shortest paths (line 8).

Algorithm 4 Aging aware routing algorithm

```
Require: Src-Dest pair list $\{(Src, Dest)\}$, Router's age list $\{RAg\}$
Ensure: List of shortest paths $\{ShortPathPair\}$
 1: ShortPathPair ={};
 2: for all $Pair_i \in \{(Src, Dest)\}$ do
 3:     $k\_ShortPath\{\} \leftarrow CalShortestPath(Pair_i)$; //Dijkstra's
 4: end for
 5: for all $Pair_i \in \{(Src, Dest)\}$ do
 6:     for all $Path_j \in k\_ShortPath_i$ do
 7:         if $(!MaxAgeR(Path_j, \{RAg\}) \wedge MinAge(Path_j, \{RAg\}))$ then
 8:             $ShortPathPair.Add(Path_j)$;
 9:         end if
10:     end for
11: end for
12: Return ShortPathPair;
```

4.6 Experimental Setup and Results

4.6.1 Setup

Our modeling and experiments are conducted using gem5 [22] which is an event-driven simulator that can simulate the behavior of a full system. In addition, we adopt a ruby memory model with mesh interconnect network. Furthermore, Garnet [3] network model is used with 5-stage routers that is embedded inside gem5. In order to extract power estimation results for these stages, we used Mcpat [88] for different ranges of fl and rs. HotSpot [70] is used to extract temperature maps of a router for different extracted powers. To get the router's floorplan for temperature analysis, the architecture in [1] is used. The floorplan is extracted for 45nm technology using Cadence toolchain.

For the aging model (Eq. 4.1 and Eq. 4.2), the values for C_{BTI} and C_{HCI} are chosen such that the maximum delay degradation in 3 years is 20% in worst case (transistors always ON, the maximum frequency ($\alpha \times f = 0.5$GHz) at temperature 380°Kelvin). In modeling stage, RS_{max} is assumed to be 10,000 cycles which can be counted using a 14-bit counter. In order to get the maximum number of flits, FL_{max}, we use a representative synthetic traffic patterns with flit injection rate = 0.05 for ϵ (or RS_{max}). As an observation, fl cannot exceed 2,300 flits. To confirm that 2×2 and 4×4 mesh NoC experiment are performed assuming a full system mode with SPLASH -2 [135] and similar observations are detected. As a result, we fixed our FL_{max} at 2,300 which can be counted by 12-bit counter. SPLASH-2 benchmarks are adopted for our experiments. Each run is a full-system architectural simulation of 16 cores interconnected via 4ϵ4 mesh topology. All routers can accept 16-byte flits and assume a virtual channel group that has 4 virtual channels which holds four flits. Each router in the system has 5 input ports and 5 output ports including the ones for the local processor caches. Each router is connected locally to one core with one L1 instruction cache, one L1 data cache, and one private L2 cache with sizes of 32kB, 32kB, and 16M respectively. The simulation

Table 4.1: Simulation setup

Name	Value
Frequency	1GHz
Number of cores	16 cores (X86 ISA)
Main memory	512 MB
L1 icahce	32KB, 2way, 64B blocks, 4cycles, pseudo LRU
L1 dcahce	32KB, 2way, 64B blocks, 4cycles, pseudo LRU
L2 size	16MB, 64B blocks, 12 cycles
Mesh	4x4
NoC routers' flit size	16B
VC number	4
VC buffer size	4* 16B

setup is listed in Table 4.1. To evaluate the impact of our proposed technique, 7 different benchmarks are selected from diverse applications. For each benchmark, we extracted their aging impact on different router in the above mentioned NoC. The results are extracted for our proposed *adaptive and online aging-aware routing algorithm* (AW) and *non-aging aware XY routing algorithm* (NAW).

4.6.2 Results

Figure 4.5 demonstrates the aging rate of the highly used router in different benchmarks and effectiveness of our proposed method. The fluctuations in AW curves (dotted-blue) demonstrate that highly aged router goes to the recovery phase (in BTI) some times. As a result, its aging rate is diminished. For example, the age of maximum aged router is improved by 45%, 42%, and 38% in FFT (router 4), Cholesky (router 0), and Radix (router 2), respectively. Second, fifth and eighth columns in Table 4.2 represents maximum age for routers in NAW, the maximum age for routers in AW, and the percentage of improvement on maximum age router for each benchmark, respectively. The maximum aged router age is improved by 39% on average. It needs to be noted that our proposed aging aware algorithm monitors aging online and reacts accordingly as opposed to previous works that determine budgets offline and through profiling for each router [75, 19, 20, 11]. While changing benchmarks and operating conditions, they have a direct impact on changing the aging rate. This may lead to timing failure because of overhead underestimation or workload's behavior overestimation.

Furthermore, in Figure 4.6 the delay degradation imbalance for different routers in the NoC is presented. As it is illustrated, in AW the routers' ages are more balanced in comparison to NAW algorithm. Hence, the load on highly aged routers are moved to lower aged routers. The age imbalance (Δ) is defined as the difference between the highly aged router (maximum) and the least aged router (minimum) in NoC. For LU_contiguous, Ocean_contiguous, and Ocean_non_contiguous benchmarks imbalance is improved by 78%, 55%, and 37% respectively. In Table 4.2 fourth, seventh, and tenth columns show an imbalance in NAW, an imbalance in AW, and percentage of improvement for different benchmarks, respectively. Our technique improves imbalance in routers' ages by 52%. Third, sixth and ninth columns in Table 4.2 are dedicated to the average aging of routers in NoC. As can be seen, our technique increases the average age of routers by 24%. The reason is that our technique balances age on all routers equally and avoid highly aged routers in the network. Therefore, the average aging of routers will increase. Our adaptive routing algorithm chooses between different shortest paths of source-destination pairs considering the age of routers across them to avoid an increase in network latency and the negative impact of performance on the system.

4.6.3 Overhead analysis

To calculate the impact of sending aging information (i.e. fl and rs) in a 4×4 mesh, 12 bits for fl and 14 bits for rs are required. Since they make a total of 26 bits, they can be encapsulated in one flit given that the number of bits per flit is 128 bits. The impact of sending that flit in traffic is estimated by counting the total number of cycles that are needed to reach router 5 from all other routers. We found out that at most 320 cycles are

Table 4.2: Aging degradation for maximum aged router, average, and its imbalance (Δ) in NoC

	NAW			Proposed method (AW)			Improvement		
	MAX (ns)	AVG (ns)	Δ (ns)	MAX (ns)	AVG (ns)	Δ (ns)	MAX (%)	AVG (%)	Δ(%)
FFT	0.095391	0.027563	0.095391	0.052433	0.034459	0.028642	45.03	-25.02	69.97
Cholskey	0.124119	0.037332	0.12418	0.072308	0.065592	0.072308	41.74	-75.69	41.77
LU_Con	0.091217	0.028434	0.087929	0.047564	0.035712	0.019708	47.85	-25.59	77.58
LU_Ncon	0.0937	0.028465	0.090576	0.048088	0.035989	0.021183	48.67	-26.43	76.61
Ocean_Con	0.09031	0.044061	0.0845528	0.071026	0.052713	0.038201	21.35	-19.63	54.81
Ocean_Ncon	0.149229	0.069564	0.136566	0.099229	0.076434	0.086566	33.5	-9.87	36.61
Radix	0.13362	0.066006	0.115422	0.082685	0.072103	0.082675	38.11	-9.23	28.37
Amean	0.111084	0.043068	0.0873868	0.067619	0.053286	0.0498976	39.12	-23.74	52.45
Gmean	0.108951	0.04016	0.074362	0.0652827	0.0505541	0.0421504	40.08	-25.88	59.2

Table 4.3: Overhead

EDP	Network Latency	Link Utilization
0.31%	0.15%	-0.25%

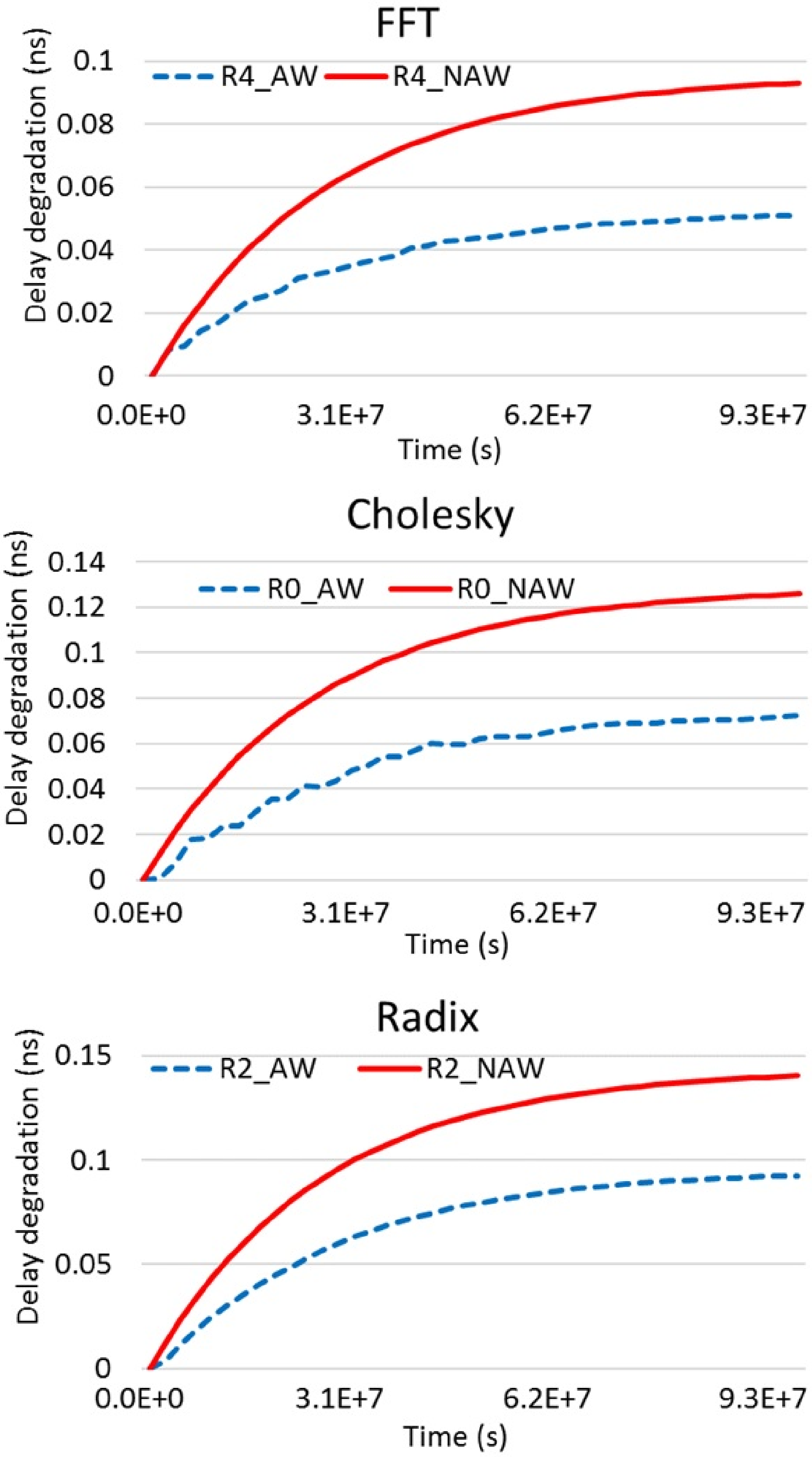


Figure 4.5: Maximum aged router comparision in 3 years (9.3E+7 seconds)

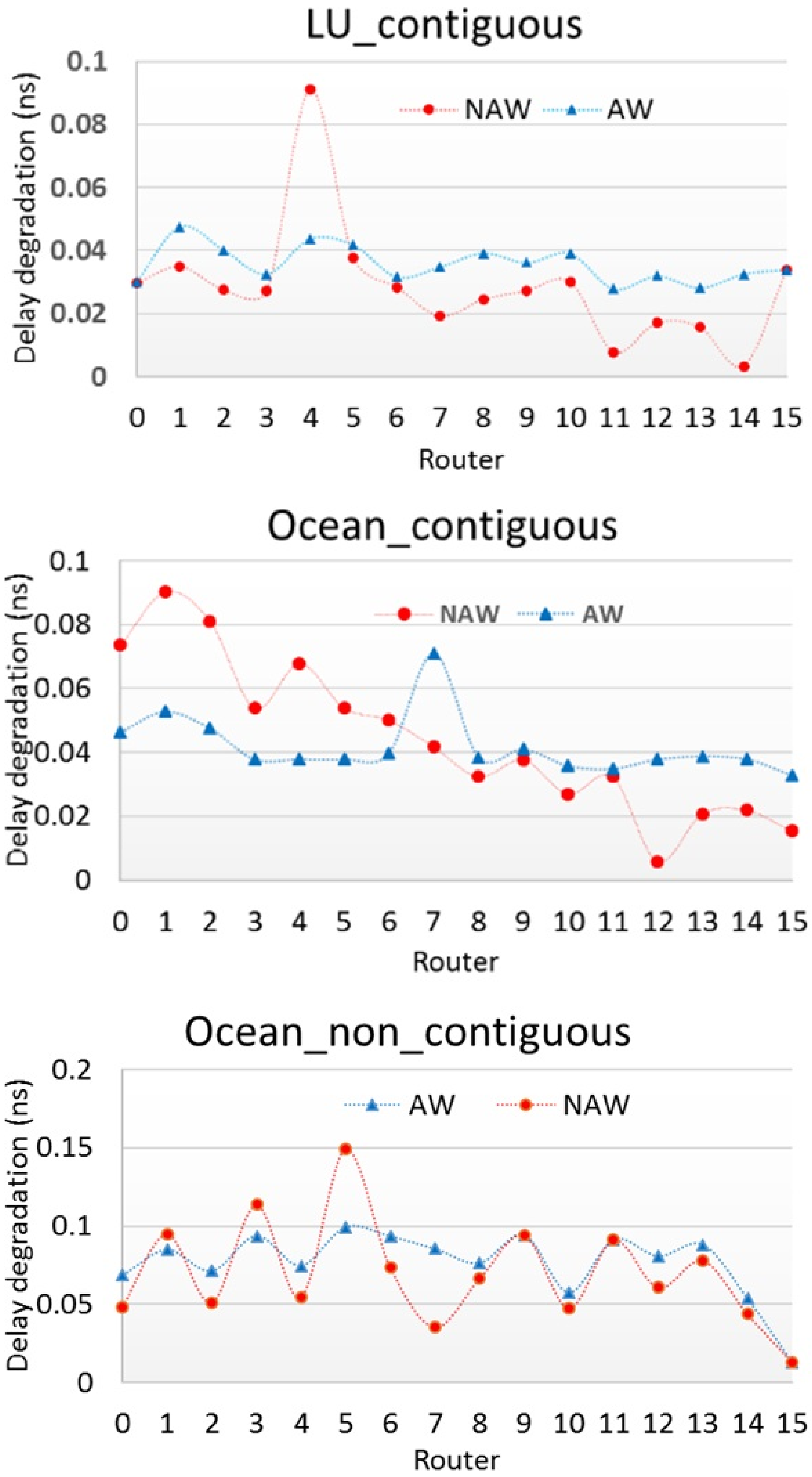


Figure 4.6: Delay degradation (age) imbalance (Δ) between different routers in NoC

needed to send all required information in each ϵ or 10,000 cycles. This account for 3.2% overhead of total traffic. We can calculate the overhead impact also by counting the number of overhead flits per router in each ϵ. The percentage of overhead flits is less than 0.2%. In addition, our method has minimal impact on EDP, network latency and link utilization, as it is demonstrated in Table 4.3. On average, EDP in AW is only 0.31% higher than NAW. Similarly, network latency is higher by 0.15% and link utilization is lower by only 0.25%.

4.7 Conclusion

In this chapter, we proposed online monitoring technique for aging in NoC routers, which is utilized in our aging-aware routing algorithm. Since routers' capacity of flits is predictable and limited in a given period of time we can predict aging rate as well. The router is analyzed for different number of flits for temperature and stress to extract a CAT. CAT is placed in one of the middle cores that has minimal distance to the other cores inside the network, which can be accessed by each NoC router based on their number of flits and resident time during a given period of time. Our experimental analysis shows 39% and 52% improvement on critical path degradation of the maximum aged router and aging imbalance, respectively, with negligible overheads.

Chapter 5

AROMa: Aging-Aware Deadlock-Free Adaptive Routing Algorithm and Online Monitoring in 3D NoCs

The movement toward 3D fabrication coupled with NoC aims to improve area, performance, power, and scalability of many-core systems. However, the reliability issue as a perpetual challenge in advanced silicon technology imperils it. Aging is an emerging reliability concern, which degrades the performance of the system and causes timing failure eventually. BTI and HCI are the dominant aging mechanisms, which escalate in higher temperature and stress (i.e. usage). In addition to the intra-layer temperature variations, 3D NoCs experience inter-layer temperature variations, which demand necessary investigations for aging as compared to 2D NoC. In this chapter, we propose AROMa, an aging-aware deadlock-free adaptive routing algorithm integrated with a novel online aging monitoring system for 3D NoCs. The monitoring system in AROMa exploits *Distributed-Centralized-Aging-Table* (D-CAT) to obtain routers' aging rates for each layer of 3D NoCs periodically. Consequently, AROMa swaps between different k-best source-destination shortest paths periodically to avoid highly aged routers, force them in the recovery phase of BTI, and accordingly balance aging in the network. We prove that AROMa is deadlock-free. Our extensive experimental analysis using gem5 full system mode for PARSEC and SPLASH-2 benchmark suites concludes that AROMa outperforms state-of-the-art works while improving age imbalance by 70% and maximum age by 35% in 3D NoC with negligible overheads.

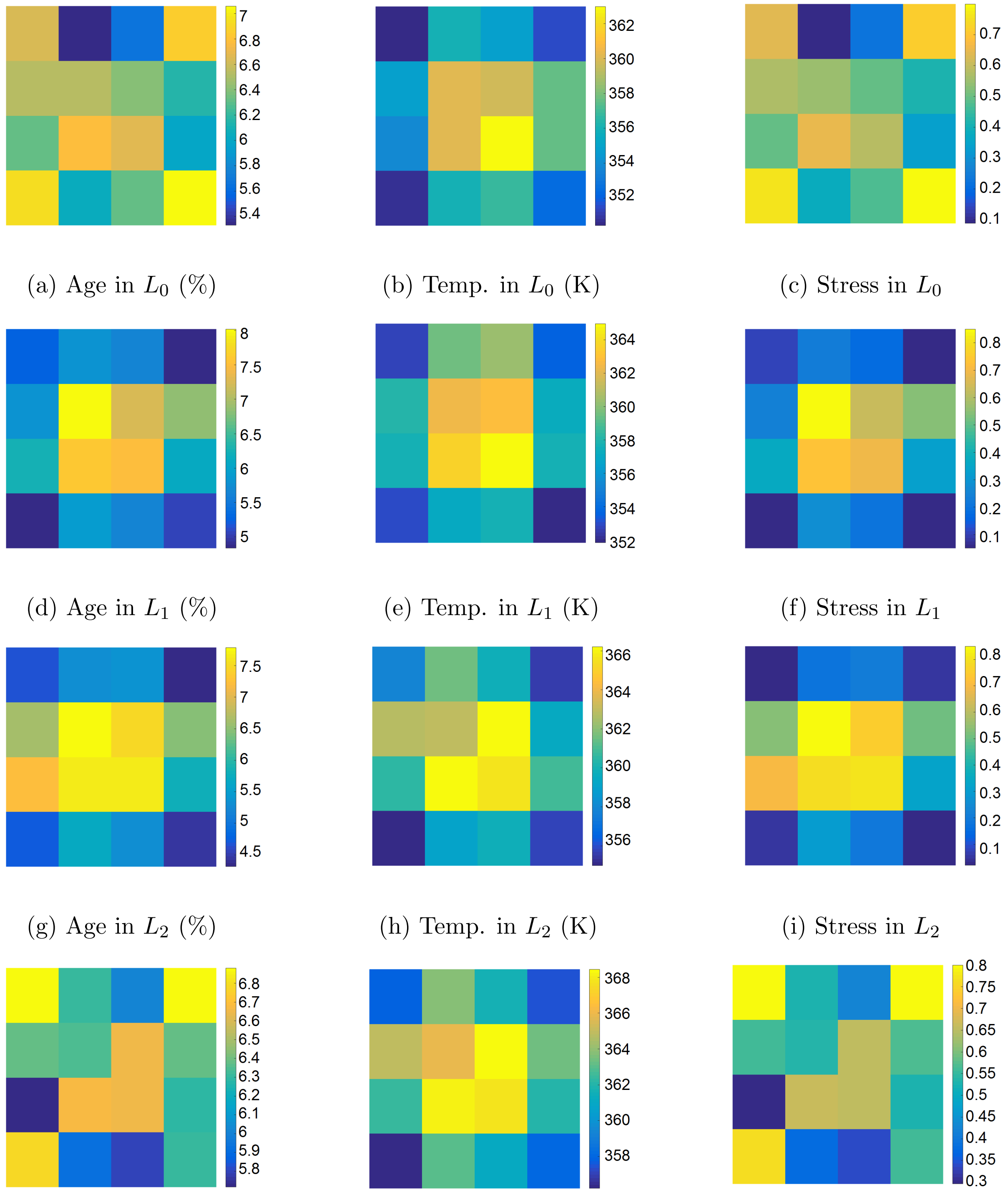
This chapter is organized as follows: an overview is briefly introduced in Section 5.1. Section 5.2 studies related work. In Section 5.3 the 3D NoC background is demonstrated. An overview of the impact of aging mechanisms in NoC is detailed in Section 5.4. Section 5.5 discusses problem formulation. AROMa aging monitoring system is proposed in Section 5.6. Section 5.7 elaborates AROMa's adaptive routing. Then, the experimental setup and results are discussed in Section 5.8. Finally, the chapter is concluded in Section 5.9.

5.1 Introduction

To cope with the complex on-chip interconnection issues in many-core systems, the three dimensional network-on-chip (3D NoC) has been proposed by applying die stacking technology for performance, energy efficiency and power consumption gains. Furthermore, 3D NoCs become a promising solution to many-core systems for its scalability which integrates a large number of homogeneous or heterogeneous intellectual properties (IP)s, e.g. processing units [5, 14, 68, 13]. Nevertheless, the reliability challenges in advanced silicon technology may jeopardize the performance gain as well as scalability of many-core systems. One of those challenges is aging mechanisms which are exasperated in high density stacked die integration, [105, 69, 43, 35].

Aging happens when a transistor is under stress and temperature is high. BTI and HCI are the dominant aging mechanisms that gradually increase the threshold voltage (V_{th}) of transistors [2, 43, 55, 58]. The shifted V_{th} leads to an undesired increase in system critical path delay which ultimately exacerbates performance loss and timing failure within the system components in the long run. Designers have to allocate considerable guardband to the critical path for avoiding timing failure. This imposes power, area, and performance overheads to the system [35, 127, 97]. Hence, NoC requires careful aging investigation to maintain high bandwidth, parallelism, and scalability in many-core systems. Aging-induced routers' performance degradation yielding to timing failure and connectivity loss in the NoC [55, 19, 20, 12, 10]. In addition, increase in temperature is currently a controversial challenge in 3D design which compels further aging investigation as compared to 2D NoC. As shown in Figure 5.1, we observed that even after running uniform random distribution of tasks, routers in different layers of a $4 \times 4 \times 4$ 3D NoC experience different temperatures and stresses. This leads to imbalanced aging degradation of routers at different layers and causes some routers to age more than the others.

Stress in BTI is the transistor's duty cycle. BTI-induced delay degradation is partially recoverable when transistor switches OFF. Stress in HCI is switching activity of transistors. Therefore, aging is a function of workload that changes both temperature and stress on the routers' critical paths' transistors. Since flits are the only stimuli in a router, both temperature as well as stress are functions of flits. Moreover, the router's capacity of flits for a given period of time is limited and also predictable. This means we can predict temperature and stress as well as the aging of a router based on flits. To this end, we count *number-of-flits* (fl) and their *residence-times* (rs) in a given period of time $t = \epsilon$. Therefore, we proposed *Centralized Aging Table* (CAT) in [55]. CAT is populated by the amount of aging degradation for different ranges of fl and rs in a router from zero up to the router's capacity for a specific time ϵ. All in all, various pairs of (fl_i, rs_j) corresponds to different temperature amounts, stress values, and thus aging rates. This allows us to monitor aging independent from the running workload.



(a) Age in L_0 (%)

(b) Temp. in L_0 (K)

(c) Stress in L_0

(d) Age in L_1 (%)

(e) Temp. in L_1 (K)

(f) Stress in L_1

(g) Age in L_2 (%)

(h) Temp. in L_2 (K)

(i) Stress in L_2

(j) Age in L_3 (%)

(k) Temp. in L_3 (K)

(l) Stress in L_3

Figure 5.1: Age, temperature (Temp.) and stress maps in each layer L_i of 3D NoC ($4 \times 4 \times 4$) for uniform random distribution.

The inter-layer temperature difference of stacked die causes imbalanced aging between them (Figure 5.1). In this work, we propose *Distributed CAT* (D-CAT) to quantify the gap induced by temperature change in different perpendicular distances of layers from the heat sink. Each layer has its own D-CAT to increase the aging monitoring system scalability and catch the temperature variation of different layers. This means that the same pair of (fl_i, rs_j) results in different aging rates because of varying temperature amounts of layers in 3D NoCs. Our proposed aging monitoring system is independent of the diverse behavior of running application and is able to monitor aging online for routers at different layers of the network.

In addition, because fl and rs pairs are stimulated by the network routing algorithm, routers' ages also have a direct correlation to their parameter. Since there are different shortest paths between source-destination pairs in an NoC, the network can adaptively swap between them to decrease the stress and temperature on the routers. We proposed AROMa, which is an aging-aware adaptive routing coupled with our novel online monitoring system to avoid aging imbalance in routers and ultimately increase the lifetime of the system.

The techniques in [19, 20, 131, 11] focused on BTI-induced aging delay degradation in 2D NoC. These techniques are either offline [19, 20], while aging is significantly affected by run-time operation conditions, or impose large overhead to the systems [131, 11]. The proposed techniques in [19, 20, 131] assign lifetime budget to each router that results in their premature routers failure. Starting with unchanged biased budgets assignment to each router incurs imbalance and unfair aging which results in over-aged routers. Furthermore, the proposed budgeting techniques in [19, 20] are dependent on the benchmark characteristic used in the profiling phase which reduces their applicability. Additionally, complex circuitry is required to compute budgets in [131] as well as a parallel network to propagate budgets in the network that sustain significant overhead. We implemented the proposed **OF**fline budgeting for adaptive **A**ging-aware **R**outing (OFAR) method based on [19, 20, 131], which assigns lower budgets to highly utilized routers during profiling to lower their load at runtime and utilized our online monitoring system for tracking the aging rates of routers.

The main contributions of this work are summarized as follows:

- We formulate aging effects caused by HCI and BTI for 3D NoCs using our proposed online monitoring D-CAT in AROMa. D-CAT is able to quantify the gap imposed by temperature change in different perpendicular distances of layers from the heat sink and system's conditions expressed by stress in 3D NoC. Using D-CAT, routers are able to keep track of their ages at each determined time ϵ.
- We proposed AROMa, an online adaptive aging-aware routing algorithm and online monitoring system for 3D NoCs. AROMa chooses one of k-best shortest paths between each source-destination pairs, which has least aged routers by avoiding the maximum aged ones. This adaptivity happens at each period of time $P = n \times \epsilon$.
- We proved AROMa is a deadlock-free technique.
- We implemented AROMa using gem5 full system mode and compare it to OFAR and **N**on-**A**ging a**W**are routing (NAW) for both 2D and 3D NoCs.

Our extensive experimental analysis using gem5 full system mode for PARSEC and SPLASH-2 benchmark suites is done for both 2D (4×8) mesh topology and its respective 3D NoCs ($4 \times 4 \times 2$) version. These results for three years of execution show that AROMa outperforms state-of-the-art works (OFAR) when compared to non-aging aware XY and XYZ routing (NAW). On average, AROMa improves maximum aging by 33% and 34% in 2D and 3D NoC, respectively, in comparison to NAW while OFAR worsens it by 31% and 51%. Similarly, AROMa improves age imbalance significantly by 61% and 72% in 2D and 3D NoC, respectively, while in OFAR age imbalance is worsened by 69% and 120%. We can conclude that since OFAR assigns budgets offline, it is not able to adaptively change between shortest paths. Also, OFAR's main purpose is to transfer traffic to less loaded routers and over-utilize these routers to the point that some of them fail. Although OFAR shows acceptable improvements for certain benchmarks, it fails on others. This shows that the previous techniques are application dependent and less flexible. Moreover, the experimental results show that 3D NoCs are more robust against aging as compared to 2D NoCs since the paths are shorter and routers experience less stress. AROMa imposes negligible overheads in comparison to OFAR.

5.2 Related work

The move from 2D to 3D NoC paradigm introduces a methodology for integrating a very high number of logic in a single die. The achievable performance benefit arising out of adopting 3D NoCs includes performance gain, functionality, and packaging density compared to 2D implementation especially in terms of throughput, latency, energy dissipation, and wiring area overhead [52]. The work in [110] also shows that through both analysis and simulation, 3D NoCs achieve better average performance.

The routing algorithm's function is to forward the flit that arrives at an input port of a router to one of its output ports. There are numerous routing algorithms for 2D [23, 101] and 3D NoCs [78], each one leads to different performance and cost. Routing algorithms can have three major criteria: decision location (source or distributed routing), the path length (minimal or non-minimal routing), and path definition (deterministic or adaptive routing). In source routing, the complete path is decided at the router connected to the source [25], while in distributed routing each router receives, stores and then defines the direction of a flit [114]. Therefore, source routing requires full knowledge about the network, which results in area and traffic overheads in routing tables and network. On the other hand, routing decision in distributed routing is spread among routers which impose less area and power overheads. In minimal routing the shortest path from source to destination is greedily chosen, e.g. *Breadth First Search* (BFS), Dijkstra, and Floyd-Warshall algorithms [39]. In contrast, non-minimal routing algorithms allow flits to traverse longer source-destination pair distance to meet other network objectives, e.g. aging or congestion avoidance [46, 125].

In deterministic routing, the path is completely specified based on the position of source and destination offline, e.g. XY routing [23]. In adaptive routing, the path is a function of online network variations, e.g. traffic in turn model [60]. Deterministic routing is easy to implement and imposes less area overhead while flits blindly follow the same path without considering the path congestion or age. In adaptive routing, we can enhance the routing to be more intelligent to choose between different shortest-paths and guarantee certain objectives. We review plethora of proposed adaptive routing algorithms in the following subsections that target different objectives such as performance, fault tolerance, congestion avoidances, load balancing, and aging mitigation.

5.2.1 Congestion aware adaptive routings

Table based adaptive routing is one of the methods to ensure a high degree of adaptivity for better performance [94, 103, 125, 46, 64]. In [103], authors propose a compression technique using graph coloring to shrink the large routing table size. [94] proposes a routing algorithm based on partitioning routers into different regions that could be accessed only through certain routers. HARAQ utilizes Q-learning method to provide alternative paths between each source-destination pairs using Q-tables in each router for local and global congestion information [46]. Additionally, [64] proposes *Region Congestion Awareness* (RCA) technique to improve global network balance using a monitoring network to estimate congestion. The monitoring network added circuitry to each router to aggregate and propagate congestion information to other routers. A channel pressure model is adopted in [125] to quantify and predicts traffic in order to implement an offline method for designing deadlock-free adaptive minimal routing to address local congestion. In all, considering only local congestion information may not choose less congested paths and lead to excessive congestions in other parts of the NoC. However, if global information is considered area and power overheads are imposed because of larger routing tables. In addition, congestion information broadcasting imposes traffic overhead.

5.2.2 Fault tolerant adaptive routings

Adaptive routing algorithms have been implemented for reliability and fault tolerance purposes in NoC [53, 83, 117]. In [53], a routing algorithm for 2D mesh and torus topologies reconfigures the routing table of each router to avoid faulty components, but they need a fixed amount of hardware per router to achieve that. The authors in [83] proposes an adaptive routing which forces the traffic to be distributed across the whole network. The algorithm distributes traffic uniformly to avoid overloading the links and faulty routers in case of failure. Furthermore, [117] aims to provide connectivity of 2D mesh NoCs even after some network components are out of service by deflective routing using a router design based on nostrum architecture. They use fine grain functional fault model and a methodology to diagnose and determine routers' status using *Cyclic Redundancy Checks* (CRC) hardware components. Also, the adaptive routing algorithm can employ the remaining functionality of partially defective routers. They support graceful degradation by retransmitting messages suffering from transient faults using *Error Correcting Codes* (ECC). In all, these techniques require additional hardware to the router as well as overhead bits (checksum) in each flit to implement EEC/CRC. The authors in [7] proposed a low-cost highly efficient, reliable routing for 3D mesh NoCs. They compare their work with planar adaptive routing and show that it outperforms the planner one with both synthetic and real traffic patterns.

5.2.3 Aging-aware adaptive routings and motivation

Adaptive routing can be used for aging mitigation and overcoming delay degradation effects in routers [10, 19, 20, 11, 131]. The authors in [10] propose an aging aware adaptive routing algorithm and router micro-architecture to route flits in paths that are experiencing minimum aging degradation. The routing algorithm has a shortest path selection stage and a recovery cycles insertion stage in overloaded routers. They used a circuitry through a series of delay buffers to measure a router delay degradation, which imposes considerable hardware overhead for that purpose.

Authors in [19, 20] proposed offline methods to avoid highly aged routers in 2D NoC. Deploying *Mixed Integer Linear Programming* (MILP) based on power-performance the optimized routing is obtained [19]. This technique assigns a budget to each router offline by profiling the benchmark traces. A routing algorithm finds the source-destination pairs' shortest paths considering router's budgets to mitigate aging. A similar approach is proposed in [20], where budgets are assigned offline for different epochs of time. These methods not only limits the usage of routers at runtime which can impact the system performance but also still leaves the unbalanced aging among the routers by assigning unbalanced budgets. Another important shortcoming is that aging strictly influenced by runtime variation in the workload that affects the stress and temperature on routers. While profiling and budgeting are done based on some specific benchmarks.

Exploiting architecture level criticality of flits, [11] presents a routing policy for 2D NoC with heterogeneous routers (i.e. routers are either buffered or buffer-less). Utilizing their *Wearout Monitoring System* (WMS), this method monitors aging in routers to deflect non-critical flits. Also, the large area overhead associated with complex WMS, deflecting flits degrades *Quality of Service* (QoS) at the system level. Furthermore, flits' criticality designation not only is a challenging issue but also imposes overhead to the system. Furthermore, a

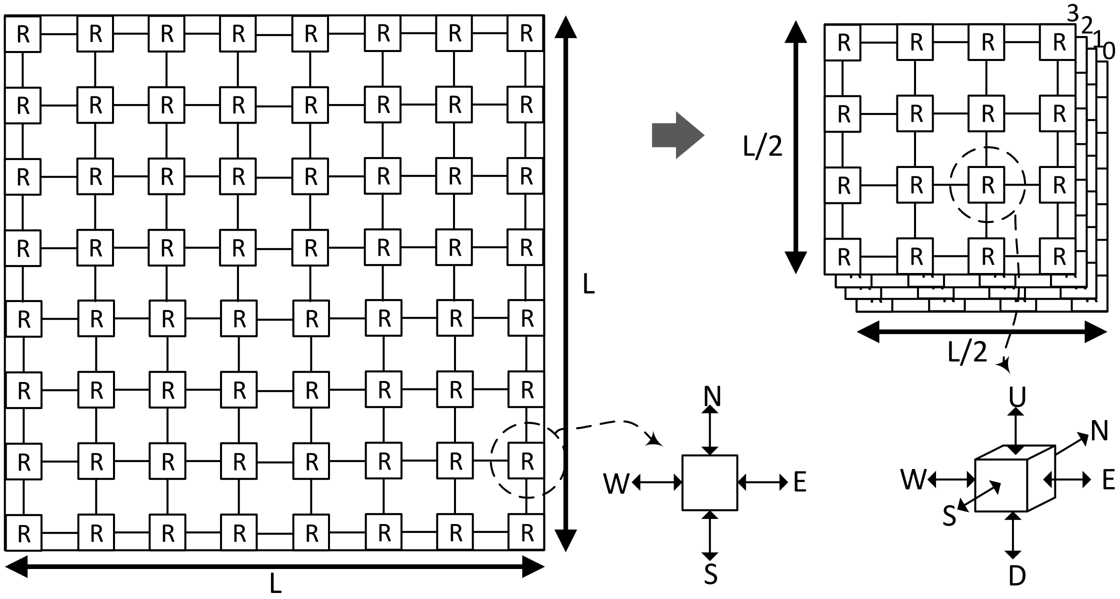


Figure 5.2: Comparison of 2D and 3D NoC area overheads.

dynamic programming based aging-aware routing is proposed in [131], which employs a lifetime budget computation unit. This technique requires a parallel dynamic programming network to propagate routers' lifetime budgets and complex circuitry for their calculations, which imposes significant overhead to the system. All the proposed methods are either offline or have a large overhead to the system. Additionally, the proposed methods are considered for 2D NoC, that can be modified for 3D NoC, but new challenges in 3D NoC such as higher temperature can impact them. In this work, we fill these gaps based on our low overhead online monitoring system that can capture workload behavior and update the routing. Authors in [54, 12] proposed aging-aware router architectures as well.

5.3 3D NoC Background

Communication between cores in systems with many cores plays a significant role [2, 66, 14]. With system design evolution throughout the decades, in order to integrate more cores in the same chip, the demand for more scalable and structural interconnect grew. Moving to NoC paradigm introduced a better alternative to the old crossbar or bus model. Therefore, several researches related to 2D NoC interconnect can be found in [68, 80].

As shown in Figure 5.2, the 2D mesh can be converted to the corresponding 3D NoC by stacking layers (4 layers in the figure) on top of each other and decrease area overhead roughly by 4× in the horizontal direction [79, 93]. The design of the 2D router has up to five ports in each direction (N, S, E, W and local). A direct extension to 3D NoC is to add two more ports in the up and down direction forming vertical links that connect different layers. These vertical links are shorter compared to horizontal links and are called *Through Silicon Vias* [137, 82]. The use of 3D mesh NoCs is intensively researched for its promising performance gain, less power consumption, reliability enhancement, design regularity, ease of implementation and heterogeneous system support compared to 2D mesh network [115, 136].

In Figure 5.3, the router architecture is depicted. We assume a pipelined router which is composed of 5 stages [3, 80, 48]. Our router [1] pipeline stages, components, and their functionalities are summarized below:

- *Buffer Write and Routing Compute* (RC): in this stage, the incoming flits are stored in input ports' buffer slots. Simultaneously, the routing-compute logic determine candidate output port and its respective virtual channel using routing table. At this stage, routing can be precomputed at the upstream router if lookahead routing is implemented.
- *Virtual Channel Allocation* (VA): its function is to assign available output virtual

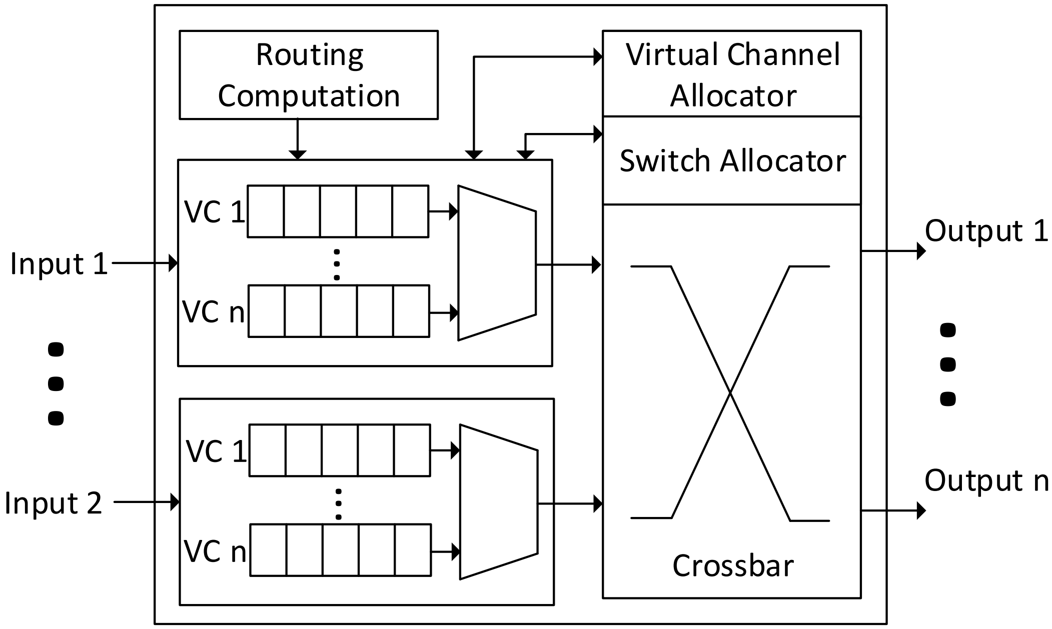


Figure 5.3: Router components and architecture.

channels to the waiting flits stored in input buffer respective virtual channel.

- *Switch Allocation* (SA): at which a flit in one of the input buffer slots which are ready to be received wins the crossbar switch time slot after arbitrating between them.
- *Switch Traversal* (ST): at this stage, flits are sent through crossbar switch to their appropriate output.
- *Link Traversal* (LT): transferring flits through links to their appropriate next router happens at this stage.

5.4 Aging-induced Delay Degradation Background

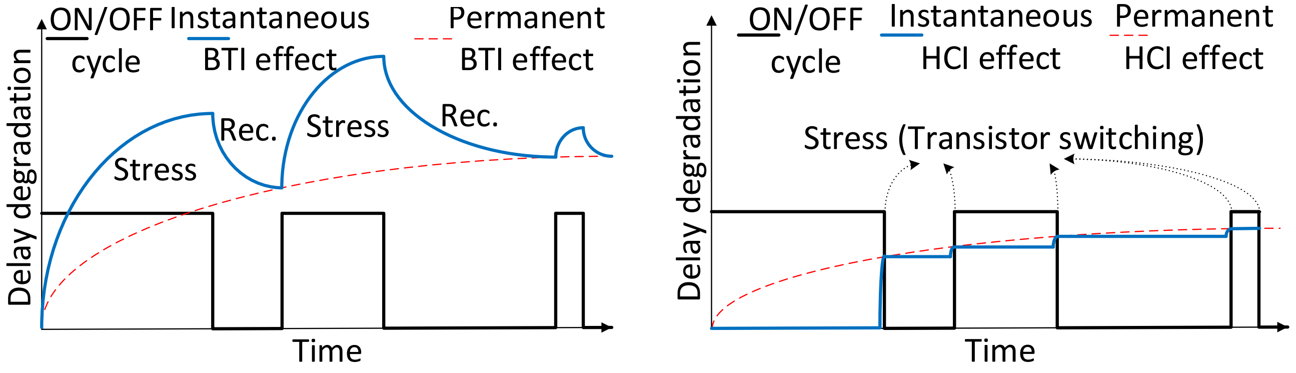
BTI and HCI are the most dominant aging mechanisms that cause aging-induced delay degradation in transistors [2, 55, 96, 56, 43]. When a transistor ages, its threshold voltage

(V_{th}) increases that leads to slower switching and higher propagation delay. Transistors aging along circuits critical paths deteriorates performance and/or causes timing failure at the system level. The delay of an aged transistor at time t can be shown as $d_t = d_0 + \Delta d(t)$, where d_0 is the intrinsic delay of transistor at time $t = 0$ and $\Delta d(t)$ is the amount of delay degradation or transistor's age. In the following subsections, we elaborate BTI and HCI aging mechanisms and their influences on the transistor's age, $\Delta d(t)$.

5.4.1 BTI aging impact

As mentioned in Subsection 4.3.1, BTI is a two-phase mechanism: stress and recovery. As shown in Figure 5.4.a, during the stress phase, the transistor is ON and its V_{th} increases gradually, while if the transistor turns OFF recovery phase starts and the threshold voltage decreases gradually but partially. During stress and recovery phases instantaneous BTI effects eventuate [127, 93] which result in permanent decay. According to [93, 126], BTI caused by interface trap generation, hole trapping in available defects, and oxide bulk trap generation. While breaking in $Si-H$ bonds at Si/SiO_2 interface results in trap generation, hole trapping in preexistent process defects is a fast phenomenon that recovers fully after stress. Whereas, oxide trap generation is dependent upon voltage drop across SiO_2 interlayer. Additional related elaborations are demonstrated in [21, 102]. This generation and destruction of traps leads to fluctuations based on stress and recovery phases. This results in an increase and then partial recovery of transistors' V_{th}. Therefore, BTI is considered a static mechanism that depends on the stress imposed by the so called duty cycle, i.e. the portion of time that transistors are ON.

The authors in [21, 102] report that the performance of transistor deteriorates caused by temperature and stress in BTI. We utilized the analytical closed form model based on them:



(a) BTI-induced delay degradation mechanism.

(b) HCI-induced delay degradation mechanism.

Figure 5.4: BTI and HCI aging mechanisms dependency on stress.

$$\Delta d_{BTI}(t) = \Delta d_{BTI}(t_{stress}) \times [1 - \sqrt{\eta \times \frac{t_{recovery}}{t}}] \tag{5.1}$$

$$\Delta d_{BTI}(t_{stress}) = C_{BTI} \times t_{stress}^{n} \times e^{(-\frac{E_a}{K \times T})} \times d_0 \tag{5.2}$$

$$t_{stress} = Y \times t \tag{5.3}$$

$$t_{recovery} = (1 - Y) \times t \tag{5.4}$$

where, T is temperature in Kelvin, Y is duty cycle, t is age in seconds, K is Boltzmann's constant, d_0 is the transistor pre-aged intrinsic delay, E_a is activation energy, n is technology dependent parameter, η is a constant and C_{BTI} is the technology dependent fitting parameter.

5.4.2 HCI aging impact

As mentioned in Subsection 4.3.2, unlike BTI, HCI is a dynamic mechanism that depends on the switching activity of the transistor, as demonstrated in Fig 5.4.b. HCI happens when accelerated electrons of the channel collide with the oxide interface and create carriers (i.e. electron-hole pairs). Some of the carriers are deposited into prohibited transistor areas (e.g. the gate oxide). As time goes on, deposited carriers alter the conductive properties of the transistor and eventually lead to increase in its V_{th}. Similar to BTI, and as depicted in Fig 5.4.b, transistor's performance degrades because of temperature and stress (switching activity) generated by HCI. We utilized the following model based on [124, 118]:

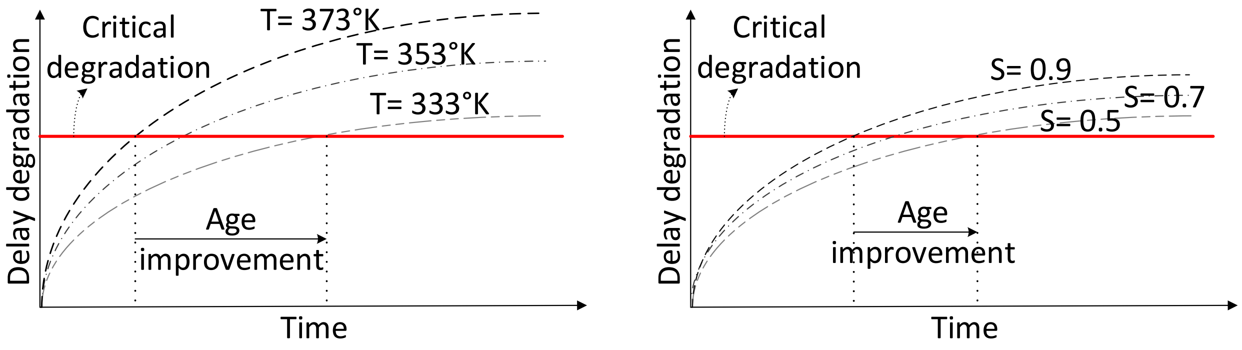
$$\Delta d_{HCI}(t) = C_{HCI} \times t_{stress} \times t^{-0.5} \times e^{(-\frac{E_a}{K \times T})} \times d_0 \tag{5.5}$$

$$t_{stress} = \alpha \times f \times t \tag{5.6}$$

where, T is temperature in Kelvin, α is switching activity, f is clock frequency in Hz, t is age in seconds, K is Boltzmann's constant, d_0 is the pre-aged intrinsic delay of transistor, E_a is activation energy, and C_{HCI} is the technology dependent fitting parameter.

5.4.3 Joint impact of BTI and HCI

As described and shown in Eq. 5.1 and Eq. 5.5, BTI and HCI aging mechanisms are exponential functions of temperature (T) and non-linear functions of stress (S). It needs to be noted that stress in BTI (S^{BTI}) is induced by duty cycle (Y) and stress in HCI (S^{BTI}) is induced by switching activity (α). Figure 5.5.a shows delay degradation (aging rate) for different temperatures but the same stress. Clearly, delay degradation commensurates with



(a) Different temperature values at same stress, S=0.5

(b) Different stress values at same temperature, T=333°K

Figure 5.5: Temperature and stress impacts on delay degradation.

higher temperature because transistors experience increasing aging rate. This results in earlier failure caused by missing the critical degradation limit (i.e the so called guardband). Similarly, in Figure 5.5.b, aging rate increases in proportion with higher stress but the same temperature. Based on Figure 5.5, we can conclude that different temperature and stress pairs leads to dissimilar aging rates. Accordingly, the transistor's age at time t for BTI and HCI aging mechanisms is equal to:

$$d(t) = d_0 + \Delta d_{BTI}(t) + \Delta d_{HCI}(t) \tag{5.7}$$

Since the aforementioned modelings compute delay degradation for time t, it is required to keep track of the aging history if we want to compute delay degradation in consecutive periods of time. For example, in Figure 5.6 it is shown that since in each time period t_{i-1} to t_i the running workload behavior and characteristics change, it will consequently affect the temperature and stress as well. For instance, during the first time period t_0 to t_1, temperature and stress pair is $< T_1, S_1 >$, while in time period t_1 to t_2, it is $< T_2, S_2 >$. Eq. 5.1 and Eq. 5.5 for time period t_0 till t_i can be rewritten as Eq. 5.8 and Eq. 5.9, respectively:

$$\Delta d_{BTI}(t_i) = \Delta d_{BTI}(T_i, S_i^{BTI}, t_i) \tag{5.8}$$

$$\Delta d_{HCI}(t_i) = \Delta d_{HCI}(T_i, S_i^{HCI}, t_i) \tag{5.9}$$

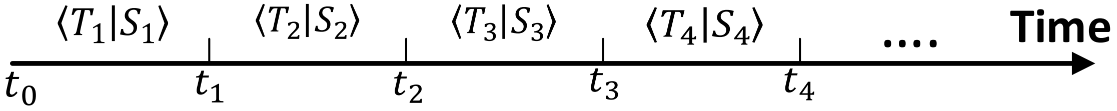


Figure 5.6: Different delay degradation associated with temperature and stress in consecutive time periods.

Utilizing Eq. 5.8, BTI-induced delay degradation can be computed for time period t_i to t_{i+1} based on delay degradation for time period t_0 to t_i as follows:

$$\begin{aligned}\Delta d_{BTI}(t_{i+1}) = \Delta d_{BTI}(&\frac{t_i}{t_{i+1}} \times T_i + \frac{t_{i+1} - t_i}{t_{i+1}} \times T_{i+1},\\ &\frac{t_i}{t_{i+1}} \times S_i^{BTI} + \frac{t_{i+1} - t_i}{t_{i+1}} \times S_{i+1}^{BTI}, t_{i+1})\end{aligned} \tag{5.10}$$

This weighted function of temperature and stress can capture the history and BTI aging recovery through reduction in stress or temperature, but HCI does not have recovery phase. We can utilize Eq. 5.9 to compute HCI-induced delay degradation for time period t_i to t_{i+1} based on delay degradation for time period t_0 to t_i as follows:

$$\begin{aligned}\Delta d_{HCI}(t_{i+1}) = \Delta d_{HCI}(T_i, S_i^{HCI}, t_i)+\\ \Delta d_{HCI}(T_{i+1}, S_{i+1}^{HCI}, t_{i+1}) - \Delta d_{HCI}(T_{i+1}, S_{i+1}^{HCI}, t_i)\end{aligned} \tag{5.11}$$

In all,

$$\Delta d(t_i) = \Delta d_{HCI}(T_i, S_i^{HCI}, t_i) + \Delta d_{BTI}(T_i, S_i^{BTI}, t_i) \tag{5.12}$$

Since BTI is the dominant aging mechanism and almost is three times higher than HCI [17, 58, 99], we define S_i as:

$$S_i = m_1 \times S_i^{BTI} + m_2 \times S_i^{HCI} \tag{5.13}$$

where m_1 is three times greater than m_2. From Eq. 5.12 and Eq. 5.13:

$$\Delta d(t_i) = \Delta d(T_i, S_i, t_i) \tag{5.14}$$

Eq. 5.14 concludes that aging is a function of stress and temperature. By finding these two characteristics of a router till time t_i, the aging rate can be predicted. Designers assign aging guardband on critical paths by predicting the worst case scenario which imposes performance, area and power overhead to the system [58, 35, 127]. Guardbands on critical paths also are added because of process variation, voltage droop, and temperature. The focus of this work is to increases the lifetime of the NoC by preserving *aging guardband* after chip fabrication and during runtime.

5.5 Problem Formulation

The objective of AROMa is to find a set of source-destination paths that satisfies the performance requirement while minimizing the maximum aged router's age. This balances the age across all routers for 3D NoCs. Hence, given:

- The list of routers $R = \{r_0, r_1, ..., r_{n-1}\}$
- The list of routers' ages $RAg^{t_i} = \{RAg_0^{t_i}, RAg_1^{t_i}, ..., RAg_x^{t_i}, ..., RAg_{n-1}^{t_i}\}$ where $RAg_x^{t_i}$ is the age of router x at time t_i.
- The list of k-best shortest-paths $KSP = \{\{KSP_{0,1}\}, ..., \{KSP_{x,z}\}, ..., \{KSP_{n-2,n-1}\}\}$

where $KSP_{x,z}$ is the list of k-best shortest-paths between source router r_x and destination router r_z.

the objective is:

$$\begin{aligned} &\text{MinMax} \quad && \mathrm{RAg_x^{t_i}}, \forall r_x \in R, \\ &\text{subject to} \quad && RAg_x^{t_i} < RAg_{GB}, \forall r_x \in R, \\ & && FindSP(KSP, RAg^{t_i}), \forall r_x, r_z \in R, \end{aligned} \tag{5.15}$$

where, RAg_{GB} is the critical path aging guardband, $FindSP(KSP, RAg^{t_i})$ is our proposed function to find shortest-paths for each pair of r_x and r_z considering the routers ages belonging to paths between them at time t_i from the list of KSP.

5.6 Online Aging Monitoring in 3D NoC

We elaborated in Section 5.4 and concluded in Eq. 5.12 that BTI and HCI are functions of temperature and stress. In addition, the only stimuli in a router, as a system, is flits. Hence, temperature and stress are functions of flits. From system point of view flits characteristics in a router are the *number-of-flits* (fl) and the amount of time that they reside inside a router, namely *residence-time* (rs), for a given period of time, epsilon (ϵ). These two parameters impact the amount of stress, power, and temperature of a router. For instance, if the number of flits is fl_j and their total residence time is rs_k for a given time period of i, the temperature T_i will be:

$$T_i = T(fl_j, rs_k) \tag{5.16}$$

Similarly, stress S_i for time period i is a function of fl_j and rs_k:

$$S_i = S(fl_j, rs_k) \tag{5.17}$$

Consequently, aging rate also is determined by these two parameters. Eq. 5.14 can be rewritten as:

$$\Delta d(t_i) = \Delta d(fl_j, rs_k, t_i) \tag{5.18}$$

where fl_j and rs_k are a range of numbers, not certain numbers.

If either fl_j or rs_k change, stress, temperature and aging rate change as well. Additionally, the router's capacity of flits is limited in a given period of time, which means that the maximum number of flits and their residence time cannot exceed a certain amount. Based on the NoC characteristics such as flit injection rate and topology, the maximum number of

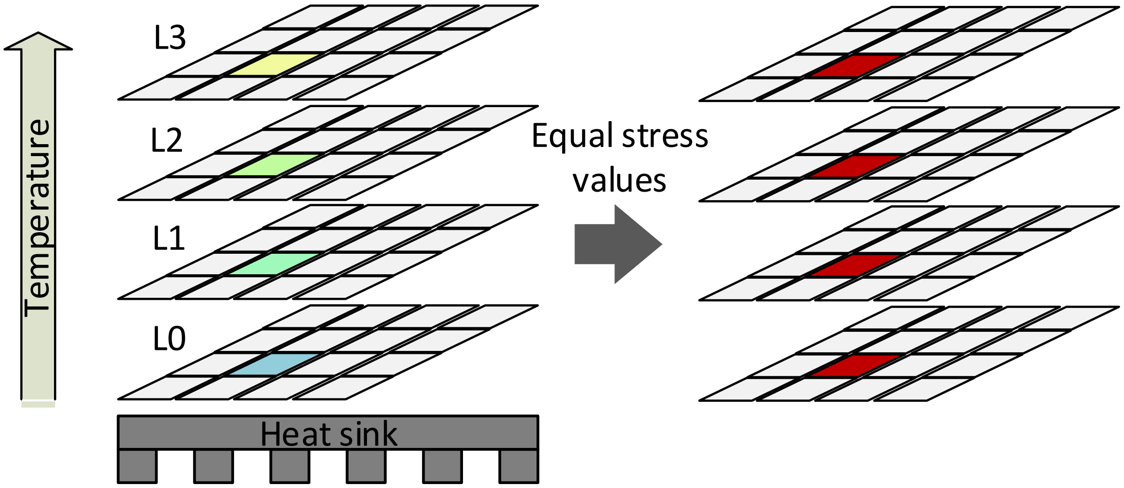


Figure 5.7: Routers in different layers of 3D NoCs with equal amount of stresses have different temperatures.

flits as well as their maximum residence time in a predetermined ϵ is bounded by FL_{max} and RS_{max}, respectively. Using Eq. 5.18, we can predict delay degradation of router x, $\Delta d_x(t_i)$, by monitoring its fl and rs during the time period of t_i. The age of router till time t_i is computed as:

$$RAg_x^{t_i} = \sum_{0}^{i} \Delta d_x(t_i) \tag{5.19}$$

Additionally, in 3D NoC, it is essential to distinguish between layers in terms of temperature because distinct layers have different temperature maps. Usually, the bottom layer that is next to the heat sink experiences a lower range of temperature than the upper layers assuming homogeneous NoC nodes. in other words, the farther distance from the heat sink the higher temperature range. Figure 5.7 illustrates that routers with the same position in each layer have different temperatures even though their stress values are equal. For instance, the average temperature difference between different layers is approximately 2 Kelvin [31]. Consequently, the same fl and rs pair values correspond to different temperatures for each layer.

For example, in Figure 5.8, it is illustrated that aging is monitored at each period of time P, which is divided to smaller equal periods of ϵ. Therefore, a period P is equal to $n \times \epsilon$. Assuming $RS_{max} = \epsilon$ equals to $10,000$ cycles, fl is equal to 250 and rs is equal to 0.6 of maximum residence time (RS_{max}), which is 6,000 cycles. Their pair of fl and rs, namely 250 and 6000, corresponds to a specific stress value but different temperatures for each layer in 3D NoC (Figure 5.7). Therefore, the same pair of fl and rs corresponds to the same aging rate for different routers in a layer but different aging rates for routers in different layers. For example, in layer L_2, stress is equal to H similar to other layers but temperature is equal to C and the corresponding aging is w.

Figure 5.9 illustrates our proposed architecture in AROMa for monitoring fl and rs. This

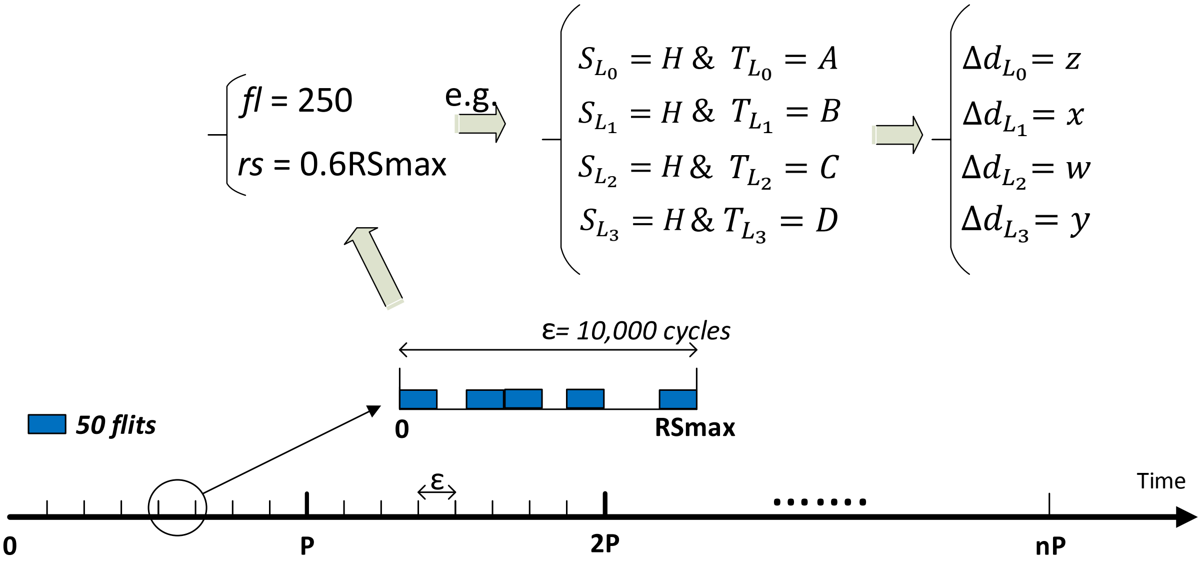


Figure 5.8: Age monitoring at each period of P at each layer L_i.

monitoring system is embedded into each router architecture [1]. Each core i is connected to a router R_i. The upper counter which is a 12-bit counter [123] counts fl for each ϵ. It monitors valid incoming flits to the router from different ports to the router using valid (v) and ready (r) signals. Therefore, whenever a flit enters a router these two signals will be active and the counter can count the number of incoming flits and find fl. This counter will be reset to zero at each ϵ (i.e. when the timer reaches ϵ).

The other parallel counter depicted in the lower section of Figure 5.9, is responsible for counting the number of cycles at which flits are residing inside a router. Basically, this counter is a timer which computes residence time, rs, of flits for each router during each ϵ. For RS_{max} or ϵ, it can be represented by 14 bits if $\epsilon = 10,000$ cycles. The counter is preceded by 14-bit subtractors. Each subtractor subtracts the exit time (Ex) of an outgoing flit, which is the current cycle when the flit exiting the router, from the en-queue time (Eq), which is saved inside the flit when it enters the router. If we assume the maximum rs of a flit inside a 5-stage router is 15 cycles, then a 4-bit MUX connected to the output of each subtractor, in order to drop any possible negative subtractions in the boundaries of each

10,000 cycles. After that, it is fed to the parallel counter to keep accumulating residence time (rs) of all flits exiting the router through all possible seven output ports (five ports in 2D architecture). Moreover, these two counters are reset after ϵ cycles. We use a timer to count ϵ and whenever it reaches to ϵ a reset signal is sent to the two parallel counters inside each router to be ready for next ϵ. The number of control signals for each counter depends on the position of the router inside the network. For instance, for a centered router in the middle layers, we have 7 ports with its corresponding control signals whereas for a router in the corner of the upper layer it has only 4 ports with fewer control signals.

To minimize the distance between D-CATs and all routers, they must be located in one of the middle routers in each layer. For example, as illustrated in Figure 5.9 D-CAT3 for layer three resides in core 53, similarly, D-CAT2 for layer two resides in core 42. For the other two bottom layers, their D-CATs reside in the corresponding cores as upper layers. The timer, which counts ϵ can be located in one of the middle layers to reset the counters inside all routers when it is required.

Based on Eq. 5.18, D-CAT in each layer will be accessed using (fl, rs) pair from all routers of that specific layer to read back their age degradation in each ϵ. Therefore, the 26 bits data (14-bit rs and 12-bit fl) is decoded to access the corresponding entry in D-CAT. Age degradation of a router can be computed for each temperature and stress (Eq. 5.10 and Eq. 5.11). To this end, we determine conditions that may happen to a router. Each condition, $C_{i,j}$, is represented by its respective rs_i and fl_j. Each pair of (rs_i, fl_j) corresponds to temperature $T_{i,j}$ and stress $S_{i,j}$ (i.e. $(T_{i,j}, S_{i,j})$). Each condition is a function of rs_i and fl_j and each condition corresponds to a specific aging rate. For example, in Figure 5.10 for layer L_0, when a number of flits is fl_j and they reside inside the router for queuing, processing and traversal through the router for rs_i cycles out of ϵ cycles, the delay degradation is $\Delta d^0_{i,j}$. Same number of flits fl_j and residence time rs_i in layer L_k leads to $\Delta d^k_{i,j}$ delay degradation. This is because the temperature varies at different layer as discussed earlier (Figure 5.7).

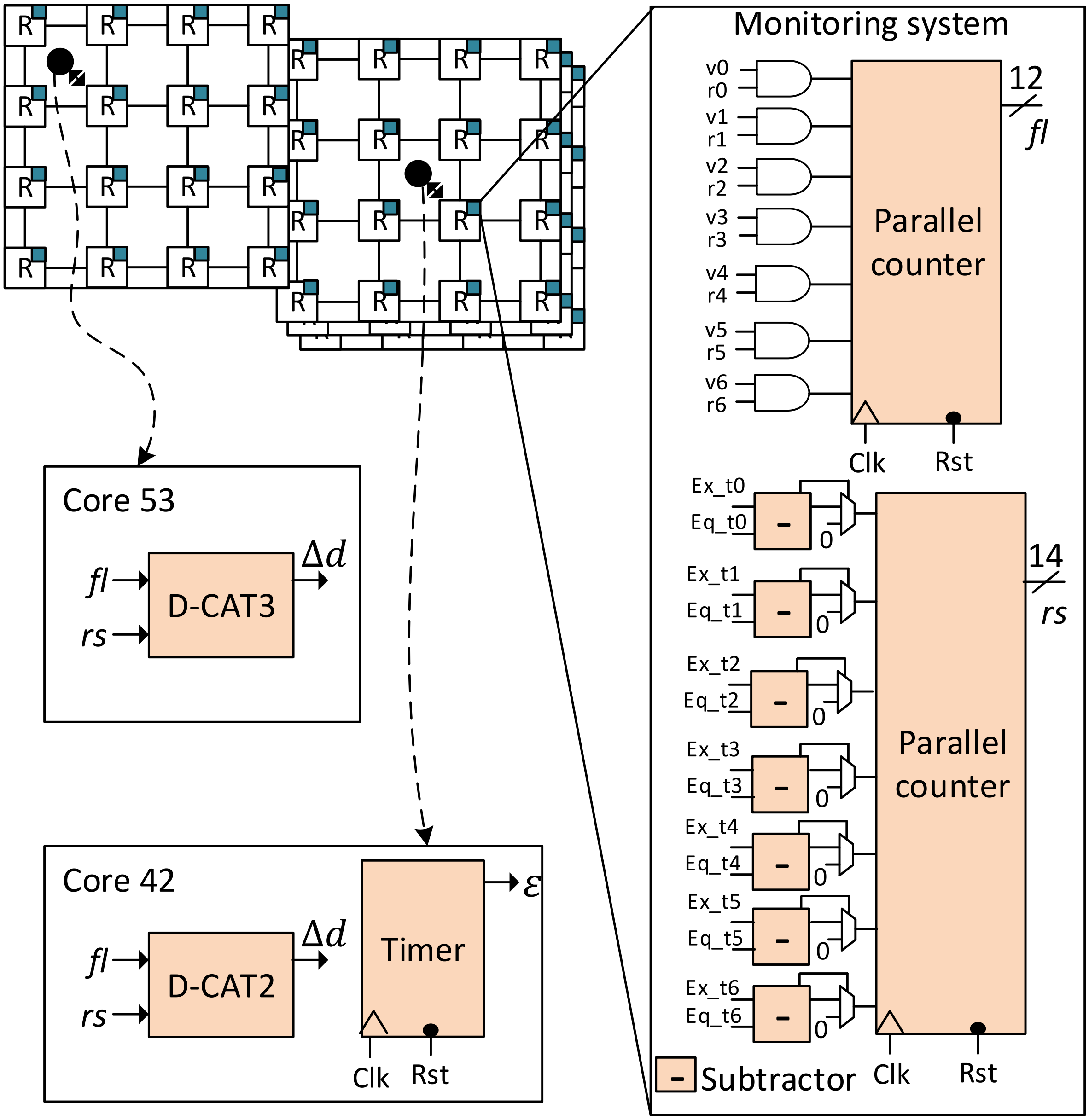
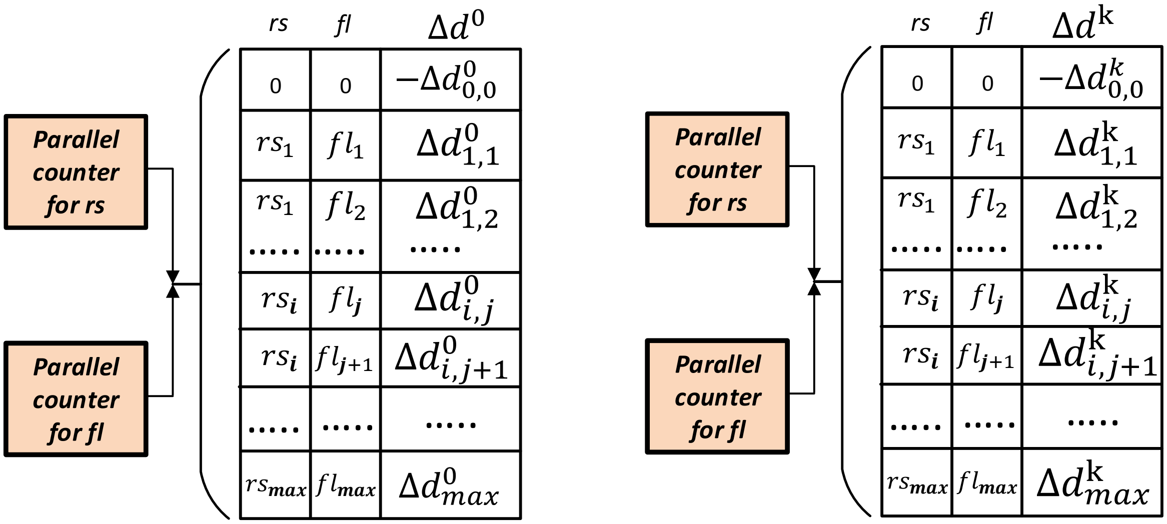


Figure 5.9: AROMa online aging monitoring architecture. Other cores (black circles) and two D-CATs are not shown for clarity).

It must be noted that when the router is not busy (fl and rs are equal to zero) and BTI recovery phase happens, D-CATs returns a negative corresponding amount of recovery as stated in their first entry.



rs	*fl*	Δd^0
0	0	$-\Delta d^0_{0,0}$
rs_1	fl_1	$\Delta d^0_{1,1}$
rs_1	fl_2	$\Delta d^0_{1,2}$ .....
rs_i	fl_j	$\Delta d^0_{i,j}$
rs_i	fl_{j+1}	$\Delta d^0_{i,j+1}$
.....	.....	.....
rs_{max}	fl_{max}	Δd^0_{max}

rs	*fl*	Δd^{k}
0	0	$-\Delta d^k_{0,0}$
rs_1	fl_1	$\Delta d^{\mathrm{k}}_{1,1}$
rs_1	fl_2	$\Delta d^{\mathrm{k}}_{1,2}$ .....
rs_i	fl_j	$\Delta d^{\mathrm{k}}_{i,j}$
rs_i	fl_{j+1}	$\Delta d^{\mathrm{k}}_{i,j+1}$
.....	.....	.....
rs_{max}	fl_{max}	$\Delta d^{\mathrm{k}}_{max}$

Figure 5.10: D-CATs for layer L_0 and layer L_k.

Algorithm 5 D-CAT construction

```
Require: Maximum resident time $RS_{max}$, number of resident time steps $rs_{steps}$, number of
    flits steps $fl_{steps}$, injection rate $Ijrate$, number of layers $L$, Router floorplan $RFLP$
Ensure: List of {D-CAT}
$FL_{max} \leftarrow FindMaxFlit(Ijrate, RS_{max})$;
$\{rs\} \leftarrow CreateRsList(RS_{max}, rs_{steps})$;
$\{fl\} \leftarrow CreateFlList(FL_{max}, fl_{steps})$;
for all $l_k \in L$ do
    for all $rs_i \in rs$ do
        for all $fl_j \in fl$ do
            $P^k_{(i,j)} \leftarrow CalPower(rs_i, fl_j)$;
            $T^k_{(i,j)} \leftarrow CalTempreture(P^k_{(i,j)}, RFLP)$;
            $S_{(i,j)} \leftarrow CalStress(rs_i, fl_j)$;
            $\Delta d^k_{(i,j)} \leftarrow CalDelayDeg(T^k_{(i,j)}, S_{(i,j)})$;
            D-CAT$^k \leftarrow FillCAT(rs_i, fl_j, \Delta d^k_{(i,j)})$;
            D-CAT$list.Add$(D-CAT$^k$);
        end for
    end for
end for
Return {D-CAT};
```

5.6.1 D-CAT construction algorithm

The pseudo code for constructing D-CATs is shown in Algorithm 5. The inputs to this algorithm are maximum residence time RS_{max} (or the updating time period (ϵ)), the steps for each residence time rs_{steps}, the number of steps for counting flits inside the router fl_{steps}, the injection rate to the system $Ijrate$, the number of layers in the 3D NoC L, and the router's floorplan $RFLP$. The algorithm's output is the list of D-CATs which are found for each layer in the 3D NoC and can be accessed from each router of its corresponding layer to read back its own age based on fl and rs during each ϵ.

In the beginning, the maximum number of flits (FL_{max}) that can occupy a router during RS_{max} (or ϵ) considering the maximum $Ijrate$ is extracted (line 1). After that, the list of residence time (rs) and number of flits (fl) will be quantized based on their number of steps (line 2, 3). As we discussed, each layer requires different D-CATs. Therefore, in a loop for each layer l_k, each different residence time rs_i and number of flits fl_j, we have different power maps ($P^k_{i,j}$). We calculate power consumption using Mcpat [88] (line 7) for each pair of rs_i, fl_j values. Consequently, different temperature maps ($T^k_{i,j}$) can be extracted for each layer k using the HotSpot tool [70]. HotSpot takes the corresponding power consumption for rs_i, fl_j, and routers' floorplan $RFLP$ as inputs (line 8). Similarly, the stress will be extracted as $S_{i,j}$ based on HCI and BTI aging mechanism (line 9). As shown in Eq. 5.3 and Eq. 5.6, Stress (S) is a function of duty cycle (Y) in BTI and switching activity (α) multiplied by clock frequency (f) in HCI. In this work, Eq. 5.13 is utilized to calculate S. For BTI mechanism, Y is equal to the residence time rs and for HCI mechanism α is equal to the ratio between fl and FL_{max}. The delay degradation is extracted for each temperature and stress pair using Eq. 5.14 (line 10). After that, the D-CAT for each layer k will be filled for each pair of rs_i and fl_j by $\Delta d^k_{i,j}$ using Eq. 5.18 (line 11). In the end, we add the D-CAT for layer k into the D-CAT list and iterate for other layers in the network till all of them in 3D NoC are covered.

5.7 Adaptive Aging-aware Routing

Imbalanced aging between routers in a network can lead to performance loss or timing failure in the system. If a highly aged router fails, it impacts the scalability and reliability of the whole system. There are different shortest paths between each pair for source and destination in a network as a graph. In addition, there are alternative paths with costs which are very close to the shortest paths in term of delay. These paths use different routers to transfer flits inside the network, which means a router can be along different source-destination pairs' shortest paths.

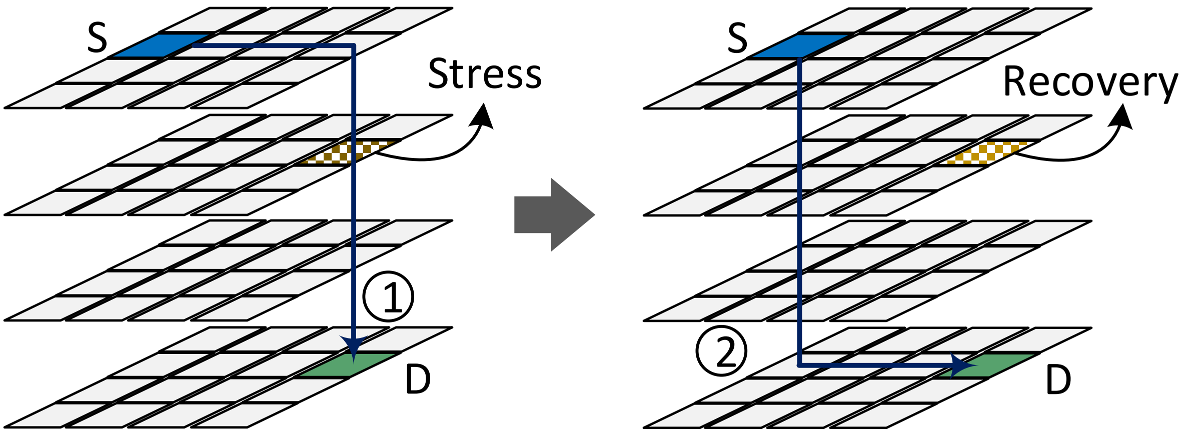


Figure 5.11: Swapping between different shortest paths that have different routers' ages.

As illustrated in Figure 5.11, two different shortest paths from source S to destination D are chosen among a set of k-best shortest paths. Each of them uses five different routers and there is no shared router between these two paths. Whereas in scenario number 1, one of the routers along the shortest paths is aged more than the others and becomes the maximum aged router. In this case, we can easily switch the shortest path to scenario number 2 without losing performance while avoiding needless increase to the age of maximum aged router and give it an opportunity to recover (i.e. in BTI). In this example, there are ten different shortest paths for the specified source-destination pairs that we can choose from for the purpose of aging mitigation.

To illustrate further, we show how routing operates as it is depicted in Figure 5.12, where a $4 \times 4 \times 4$ 3D NoC is connecting source router 0 to destination router 63. Every router maintains a routing table with size of $O(N)$ where N represents the number of routers, as it is shown in the figure for router 47. The first column represents the destination (DS) and the second one corresponds to the next router (NR). For the other routers, only the tuple where $DS = 63$ is shown. The left side of the Figure 5.12 represents the NoC at period iP. The flits follow the path defined by $0 \rightarrow 16 \rightarrow 32 \rightarrow 48 \rightarrow 52 \rightarrow 56 \rightarrow 60 \rightarrow 61 \rightarrow 62 \rightarrow 63$. After period iP is consumed and assuming routers 47 and 52 are aged more than the other routers, they are avoided in the next period. Therefore, at period $(i+1)P$, the routing tables of the routers are adjusted to new values to avoid the aged routers. At period $(i+1)P$ as shown in Figure 5.12 right side, the new path for the flits will

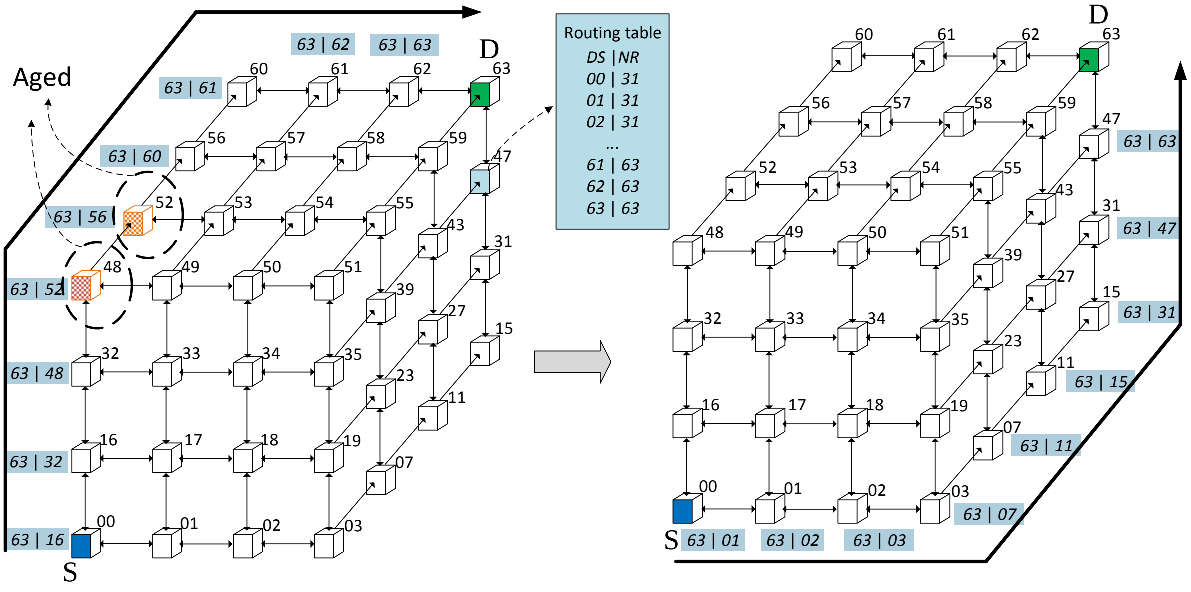


Figure 5.12: Updating routers' routing tables' entries for the new shortest path from S to D considering high aging in routers number 48 and 52. Routing table is detailed for router number 47.

be 0 → 1 → 2 → 3 → 7 → 11 → 15 → 31 → 47 → 63.

Algorithm 6 Aging aware adaptive routing algorithm

```
Require: Src-Dest pair list $\{(Src, Dest)\}$, Router's age list $\{RAg\}$
Ensure: List of shortest paths $\{ShortPathPair\}$
ShortPathPair =$\{\}$;
for all $Pair_i \in \{(Src, Dest)\}$ do
    $k\_ShortPath\{\} \leftarrow CalShortestPath(Pair_i)$;
end for
for all $Pair_i \in \{(Src, Dest)\}$ do
    for all $Path_j \in k\_ShortPath_i$ do
        if $(!MaxAgeR(Path_j, \{RAg\}) \wedge MinAge(Path_j, \{RAg\}))$ then
            $ShortPathPair.Add(Path_j)$;
        end if
    end for
end for
Return ShortPathPair;
```

5.7.1 Adaptive aging-aware routing algorithm in AROMa

Since aging is a gradual and slow mechanism, we can swap between different shortest paths in each period of time P (e.g. each week) to avoid highly aged routers in the network. The pseudo code in Algorithm 6 proposes an aging-aware routing algorithm, where we add a tag to each router as its age. This tag will be updated online using their corresponding D-CATs, periodically ($P = n \times \epsilon$). The aging tag is leveraged for choosing the best aging aware shortest path between all available k-best shortest paths from each source-destination pairs. When the new aging-aware shortest path is chosen among all k-best shortest path, the routing table in each router will be updated adaptively at each period of time P (Figure 5.12).

Inputs to Algorithm 6 are a list of source-destination pairs, $\{(Src, Dest)\}$ and list of routers' age, $\{RAg\}$. The algorithm's output is the list of shortest paths for each source-destination pairs, $\{ShortPathPair\}$. The routing table of each router will be updated based on new shortest paths output from Algorithm 5. Using $CalShortestPath()$ function, we find k best shortest paths list for each pair of source-destination. Dijkstra's shortest path algorithm is leveraged to find this list. There are different algorithms that can be utilized for this purpose [51, 8]. After that, for each pair we check which paths do not include the maximum aged router by calling $MaxAgeR()$ function and then finding the best paths based on the minimum summation of ages on their routers using the list of ages by calling $MinAge()$ function (line 7). The new shortest paths for each source-destination pairs are found for the next P and are added to the list of shortest paths (line 8). At the end of each period, the shortest paths are updated between all the pairs in the network using the proposed algorithm.

The age of each router is obtained from their corresponding D-CATs. After that, the list of routers' ages $\{RAg\}$ will be updated. As exemplified in Figure 5.12, the routing table are updated to swap to the new aging-aware shortest path, adaptively. In all, we utilized Dijkstra's algorithm to find k-best shortest paths by adding routers age. As formulated in Section 5.5, in Eq. 5.15, our goal is to minimize the age of maximum aged router and balance the aging among different routers in the network. It must be noted that when the highly aged router is avoided, its links also are avoided which means our algorithm inherently minimizes the age of links, as well.

Deadlock-freedom for aging-aware routing in 3D NoCs

Deadlock situations prevent flits from traversing the NoC because they are waiting for resources (i.e channels in NoCs) that are reserved by other independent requests. It occurs because head flits enter infinite cyclic waiting for specific free channels in the network that are not available. One way to ensure deadlock-freedom is to prevent circular wait for channels. As shown in Figure 5.13, there are 24 different 90-degree turns in a 3D NoC which can create 6 abstract cycles. At least one turn must be forbidden in each of these 6 abstract cycles to guarantee deadlock-freedom. There are 4096 various ways to prevent these six turns, 176 avoid deadlock and 9 are unique considering symmetries [59]. Similar approach for 2D NoCs is detailed in [60].

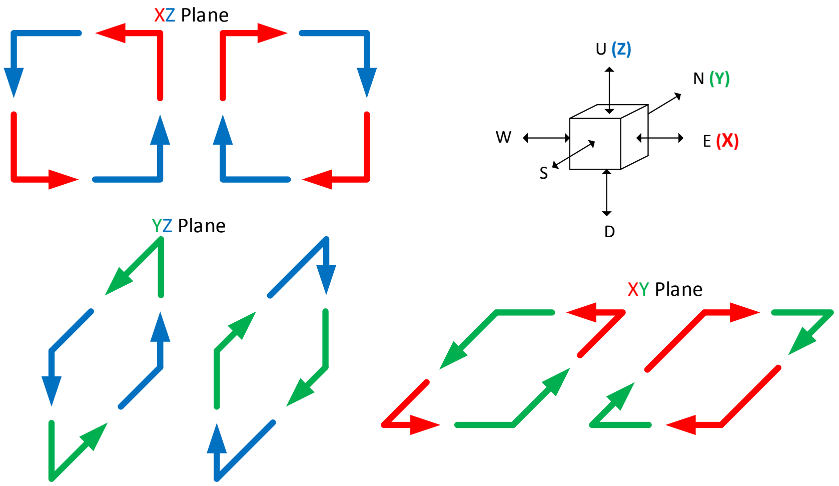
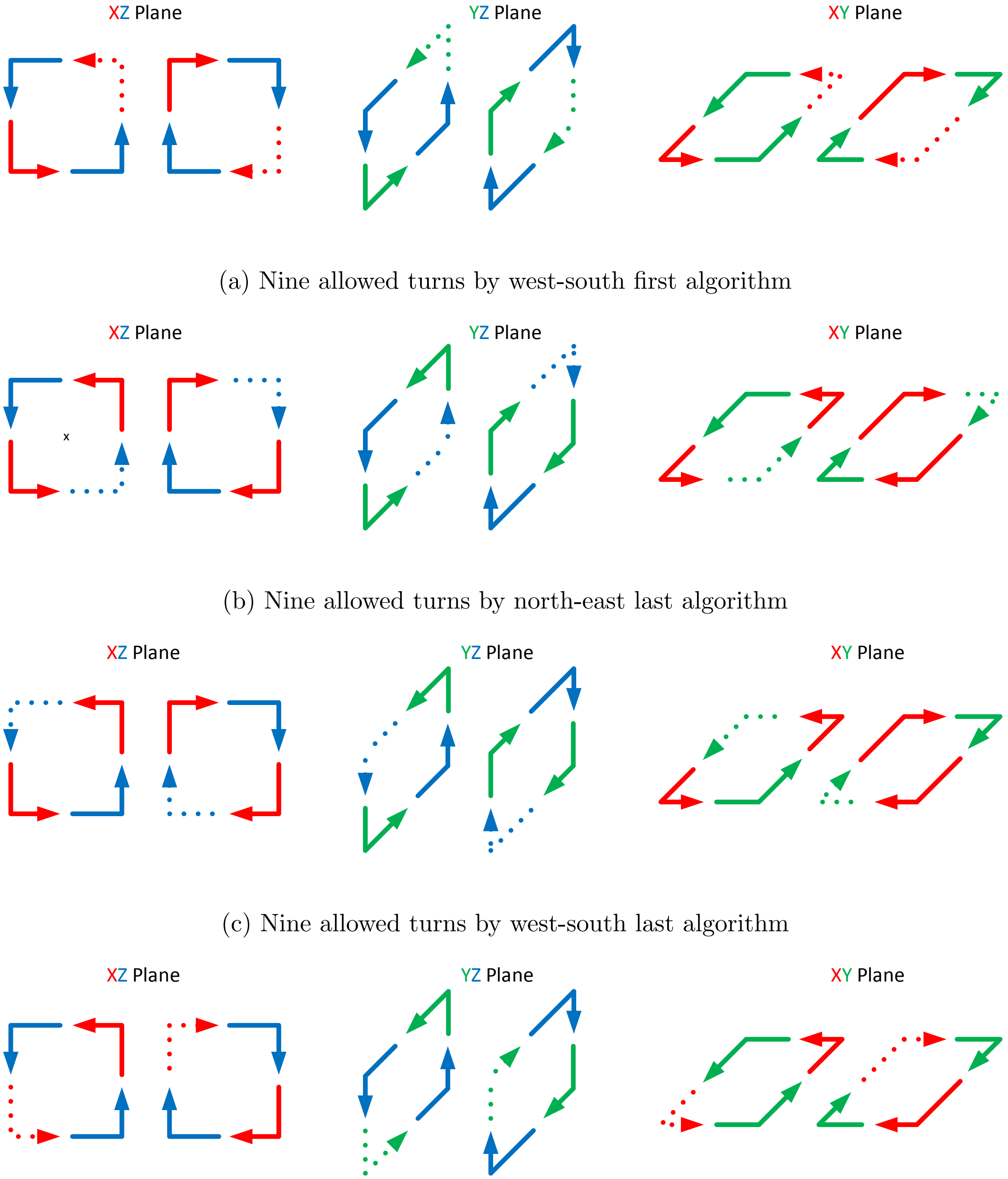


Figure 5.13: The possible abstract cycles in different planes of 3D NoCs.

Out of these 9 combinations in 3D NoCs, The following deadlock-free routing algorithms are utilized in our Algorithm 6:

- west-south first routing algorithm: where flits are adaptively routed west and south first then east, north, up, and down (Figure 5.14a).
- north-east last routing algorithm: where flits are adaptively routed up, down, south and west then they are router north and east last (Figure 5.14b).
- west-south last routing algorithm: where flits are adaptively routed up, down, east and north then they are router west and south last (Figure 5.14c).
- north-east first routing algorithm: where flits are adaptively routed north and east first then south, west, up, and down (Figure 5.14d).



(a) Nine allowed turns by west-south first algorithm

(b) Nine allowed turns by north-east last algorithm

(c) Nine allowed turns by west-south last algorithm

(d) Nine allowed turns by north-east first algorithm

Figure 5.14: Four various combinations of allowed turns (solid lines) to have deadlock-free algorithms in 3D NoCs

Deadlock-free aging-aware routing

Algorithm 6 returns a set of shortest-paths for the next period of time P that includes k-best shortest-paths available between each source and destination. Theorem 1 in [121], proves that for a 2D NoC with an arbitrary set of shortest-paths, *virtual channels* (VCs) are statically allocated to flows that ensure the network deadlock freedom as long as at least two VCs are available per port. We use this proof for 2D NoCs in our work. Similarly, we prove that a 3D NoC is deadlock-free for an arbitrary set of shortest paths (e.g. output of AROMa) as long as there are four VCs per port. These VCs are assigned statically to the four shortest-paths flow subsets detailed in Section 5.7.1.

Using the proposed turn models in Section 5.7.1, we show that any set of shortest-paths returned from Algorithm 6 can be deadlock-free through static VC allocation. The main characteristic (i.e. constraint) of a minimal path is its direction. In other words, a shortest-path flow cannot turn back toward its source and miss (lose) its minimality constraint. Considering this, in a 3D NoC:

- There are 6 different kinds of zero-turn shortest-path (i.e. +x, -x, +y, -y, +z, -z).
- There are 24 different kinds of one-turn shortest-path (i.e. (+x, +y), (+x, -y), (+x, +z), (+x, -z), ...).
- There are 72 different kinds of two-turn shortest-path (i.e. ((+x, +y), +x), ((+x, +y), -z), ((+x, +y), +z), ...).
- There are 144 different kinds of three-turn shortest-path (i.e. (((+x, +y), +x), +z), (((+x, +y), +x), -z), ...).

where + and - are the network directions, and x, y and z are its dimensions.

Theorem 1. *Given a router with at least four VCs in a 3D NoC, and an arbitrary set of shortest paths over $n \times n \times n$ network, it is possible to statically allocate VCs to each flow and ensure deadlock-freedom.*

Proof. Given the constraint of minimality, shortest-path in 3D NoCs at most have three turn types (i.e. one of the above-mentioned three-turn kinds out of 144), even though they may have many turns of the same kind. Figure 5.15 illustrates 4 out of 144 (the rest are not shown for lack of space) different three-turn shortest-path types that conform to one of the deadlock-free turn models in Section 5.7.1. We can partition and classify an arbitrary set of aging-aware shortest-paths (e.g. shortest-path generated by Algorithm 6 into four sets (i.e. a, b, c, d) then assign VC0, VC1, VC2, and VC3 of each port of routers to a, b, c, and d, respectively, to guarantee deadlock freedom in a 3D NoC. Notice that shortest paths which belong to zero-turn, one-turn, and two-turn sets can be considered as special cases of the three-turn set and assigned to one of these four subsets. This proof is an extension of the proposed proof in [121] for 2D NoCs. □

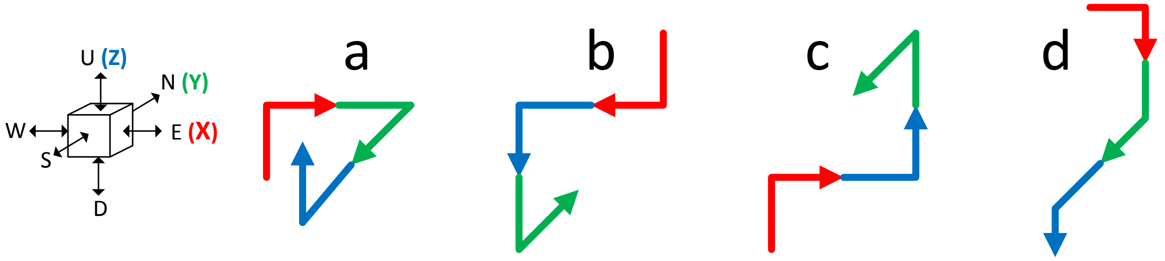


Figure 5.15: Four different three-turn shortest-path type in 3D NoCs that conform to one of the deadlock-free turn models (a, b, c, and d) in Section 5.7.1.

Additionally, authors in [45] prove that the minimum number of VCs for a fully adaptive deadlock-free routing in a 3D NoC is 16 (as opposed to 24 in our proof). However, it may eliminate some potential aging-aware shortest-paths, while our proof guarantees deadlock-free network for any set of shortest-path.

An Alternative Deadlock-freedom for 3D NoCs based on EbDa

As we stated, deadlock situations occur because head flits enter infinite cyclic waiting for specific free channels in the network but they never turn out to be available. One way to ensure deadlock-freedom is to prevent circular wait for channels. In earlier works such as [41], designing a deadlock-free algorithm depends on removing cyclic dependencies on the channel dependency graph using turn model [61]. Al alternative new methodology to ensure deadlock-freedom based on EbDa can be employed [45]. EbDa results in all shortest paths mentioned in Algorithm 6 are paths that follow turns which ensure deadlock-freedom. To ensure deadlock-freedom based on EbDa [45], three major theorems must be held. Their equivalent in 3D NoCs are as follows:

1. A partition (i.e. set of dimensions D $\subset$ X+,X-,Y+,Y-,Z+,Z-) is cycle-free if it covers at most one complete D-pair (i.e either X+,X-, Y+,Y- or Z+,Z-).
2. A partition is cycle-free if one U-turn (i.e. the transition from one channel to another in the opposite direction which called 180-degree turn) is allowed per complete D-pair taken in ascending order.
3. Transition between disjoint acyclic partitions in a consecutive order do not form a cycle.

In our work, we did not include U-turns, hence, Theorem 2 does not apply. In addition, it is also proven in EbDa that the minimum number of channels for 3D NoCs to be deadlock-free and fully adaptive is 16 for the whole router based on the formula $N = (n+1)2^{(n-1)}$ where n is the network dimension. A systematic procedure to apply EbDa in 3D is to arrange dimensions by dividing the 3D NoC into partitions with for example 3, 2 and 3 virtual channels along both negative and positive direction of X, Y, and Z dimension respectively. By applying Theorem 1 and the partitioning procedure, the partitions are generated by

Table 5.1.

Table 5.1: Generating partitions using EbDa

steps	Sets	Partitions

Form sets with all possible dimensions.
Place complete Z1 pair and one direction from the other two dimensions and remove them from sets.
	Set1:
	Dz={Z1+,Z1-,Z2+,Z2-,Z3+,Z3-}
	Set2:
	Dx={X1+,X1-,X2+,X2-,X3+,X3-}
	Set3:
	Dy={Y1+,Y1-,Y2+,Y2-}
		PA={Z1+,Z1-,X1+,Y1+}

Place complete Z2 Pair and one direction from the other two dimensions and remove them from sets.	Set1:
	Dz={Z2+,Z2-,Z3+,Z3-}
	Set2:
	Dx={X1-,X2+,X2-,X3+,X3-}
	Set3:
	Dy={Y1-,Y2+,Y2-}
		PA={Z1+,Z1-,X1+,Y1+}
		PB={Z2+,Z2-,X1- ,Y2+}

Re-order sets
(set1 and 2 are swapped)
Place complete X1 pair and one positive from each other dimension in one partition and remove them from sets.
	Set1:
	Dx={X3+,X3-}
	Set2:
	Dz={Z3-}
	Set3:
	Dy={Y2-}
		PA={Z1+,Z1-,X1+,Y1+}
		PB={Z2+,Z2-,X1- ,Y2+}
		PC={X2+,X2-,Z3+,Y1-}

Place the remaining dimensions in the last partition.	Set1:
	Dx={X3+,X3-}
	Set2:
	Dz={Z3-}
	Set3:
	Dy={Y2-}
		PA={Z1+,Z1-,X1+,Y1+}
		PB={Z2+,Z2-,X1- ,Y2+}
		PC={X2+,X2-,Z3+ ,Y1-}
		PD={X3+,X3-,Z3- ,Y2-}

	Set1: Dx={}
	Set2: Dz={}
	Set3: Dy={}
		P=
		{PA={Z1+,Z1-,X1+,Y1+};
		PB={Z2+,Z2-,X1- ,Y2+};
		PC={X2+,X2-,Z3+ ,Y1-};
		PD={X3+,X3-,Z3- ,Y2-}}

Theorem 1 allows having all possible turns within a partition while Theorem 3 allows the inclusion of the turns between disjoint partitions based on consecutive orders. Figure 5.16 explained the allowed transitions.

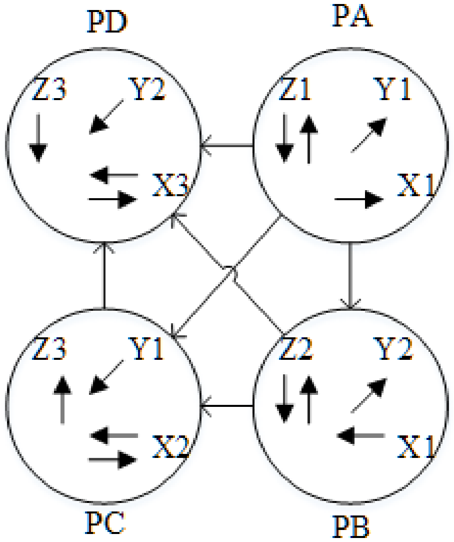


Figure 5.16: EbDa allowable transitions for 3D NoC based on the derivation from table 5.1

In Summary, the Table 5.2 described all allowable turns. Note that East (E), West (W), North (N), South (S), Up (U), Down (D) represent X+, X-, Y+, Y-, Z+, Z-, respectively and the I-turns are the transitions from one channel to another along the same direction (i.e. 0-degree turns). In addition, more VCs might be required for symmetric routing. This will increase the number of identical turns as well as U-turns and I-turns but still ensure deadlock-freedom.

5.8 Experimental Evaluation

In this section, we evaluate our methodology. First, our simulation environment setup is explained. After that, we describe our results for AROMa in 2D and 3D NoC as compared to non aging-aware (NAW) technique and state-of-the-art work. Finally, we analyze AROMa from different aspects.

Table 5.2: The restricted allowable transitions to prevent cycles and ensure deadlock-freedom using EbDa

Transition and respective theorem applied	Turns that ensure deadlock-freedom	I-turns
Within PA, Theorem 1	{E1U1, E1D1, E1N1, N1U1, N1D1, N1E1, U1E1, D1E1, U1N1, U1E1}	
Within PB, Theorem 1	{W1U2, W1D2,W1N2,N2U2,N2D2, N2W1,U2W1,U2N2,D2W1, D2N2}	
Within PC, Theorem 1	{E2U3, E2S1, W2U3, W2S1, SE2, S1W2, S1U3, U3E2, U3W2, U3S1}	
Within PD, Theorem 1	{E3D3, E3S2, W3D3, W3S2, S2E3, S2W3, S2D3, D3E3, D3W3, D3S2}	
PA to PB, Theorem 3	{E1N2, E1U2, E1D2, N1W1, N1U2, N1D2, U1W1, U1N2, D1W1, D1N2}	N1N2, U1U2, D1D2
PA to PC, Theorem 3	{E1U3, E1S1, N1E2, N1W2, N1U3, U1E2, U1W2, U1S1, D1E2, D1W2, D1S1}	E1E2, U1U3
PA to PD, Theorem 3	{E1D3, E1S2, N1E3, N1W3, N1D3, U1E3, U1W3, U1S2, D1E3, D1W3, D1S2}	E1E3, D1D3
PB to PC, Theorem 3	{W1U3, W1S1, N2E2, N2W2, N2U3, U2E2, U2W2, U2S1, D2E2, D2W2, D2S1}	W1W2, U2U3
PB to PD, Theorem 3	{W1D3, W1S2, N2E3, N2W3, N2D3, U2E3, U2W3, U2S2, D2E3, D2W3, D2S2}	W1W3, D2D3
PC to PD, Theorem 3	{E2D3, E2S2, W2D3, W2S2, S1E3, S1W3, S1D3, U3E3, U3W3, U3S2}	E2E3, W2W3, S1S2,

5.8.1 Setup

All of our simulations are done in the full system simulation mode using gem5 [22] that runs on Linux operating system to support scheduling benches for three years (9.3E+7 seconds) of execution time. In addition, we adopt a ruby memory model with 2D and 3D mesh interconnect network. Also, Garnet [3] network model is used with 5-stage routers that is embedded inside gem5. In order to extract power estimation results for these stages, we used Mcpat [88] for different ranges of fl and rs. HotSpot [70] is used to extract temperature maps of a router for different extracted powers. To get the router's floorplan for temperature analysis, the architecture in [1] is used. The floorplan is extracted for 45nm technology using Cadence toolchain.

Table 5.3: Simulation platform configuration

item	**Description**
Processor	X86 based 1.0 GHz in order cores.
L1-iCache	private, 32KB, 2-way set associative, 64B blocks, 4 cycles latency, pseudo LRU replacement.
L1-dCache	private, 32KB, 2-way set associative, 64B blocks, 4 cycles latency, pseudo LRU replacement.
L2-Cache	private, 16MB, 8-way set associative, 64B blocks, 12 cycles latency.
Main Memory	512MB. DRAM
NoC	4×8 2D mesh, $4 \times 4 \times 2$ 3D mesh, each node consists of 1 router, 1 core, 1 private L1 i/dcache, and 1 private L2 cache. MOESI cache coherence protocol, 5-stage pipeline router.
Flit size	16B
Buffer size	$4 \times 16B$ or 4 flits per virtual channel.

SPLASH-2 and PARSEC benchmarks are adopted for our experiments. Each experiment run with 32 cores interconnected via 4×8 2D mesh or $4 \times 4 \times 4$ 3D topology. All routers accept 16-byte flit sizes and assume a virtual channel architecture that has four virtual channels which holds four flits. Each 2D router in the system has five input ports for (N, E, S, W and local). 3D routers have seven input ports for (N, E, S, W and local). The local ports are connected to one core with one L1 instruction cache, one L1 data cache, and one private L2 cache with sizes of 32kB, 32kB, and 16M, respectively. Since the clock frequency is equal to 1GHz (critical path will be 1 ns) and the guadrband is chosen to be 16% (0.16 ns) for 3-year worst-case execution. Worst case happens when the temperature is 380K and the transistor is always ON. The rest of the simulation setup is listed in Table 5.3.

In modeling stage, $RS_{max} = \epsilon$ is assumed to be $10,000$ cycles which can be counted using a 14-bit parallel counter (Figure 5.9 & Figure 5.10). To get the maximum number of flits,

Table 5.4: Aging-induced delay degradation for maximum aged router and network age imbalance (Δ) in 2D NoC for 3 years of execution (9.3E+7 seconds).

	NAW		OFAR				AROMa			
Benchmark	MAX(NS)	Δ(NS)	MAX(NS)	Δ(NS)	MAX(%)	Δ(%)	MAX(NS)	Δ(NS)	MAX(%)	Δ(%)
FFT	0.101873	0.0603584	0.074661	0.036366	26.71	39.75	0.067989	0.020266	33.26	66.4
CANNEAL	0.098927	0.057665	0.070327	0.032300	28.91	43.99	0.066582	0.024100	32.67	58.21
LU_NON_CON	0.152819	0.152819	0.172457	0.172457	-12.85	-12.85	0.091096	0.076563	40.39	49.90
RADIX	0.144387	0.063254	0.395172	0.365172	-173.69	-477.31	0.098993	0.013055	31.43	79.36
SWAPTIONS	0.101935	0.060667	0.071693	0.033939	29.67	44.06	0.070238	0.026983	31.09	55.52
X264	0.109525	0.066149	0.080548	0.046149	26.46	30.23	0.072390	0.027666	33.90	58.17
BLACKSCHOLES	0.097498	0.056513	0.069047	0.031285	29.18	44.64	0.061431	0.019239	36.99	65.96
CHOLESKY	0.199071	0.147790	0.349684	0.349683	-75.66	-136.61	0.132243	0.054992	33.57	62.79
LU_CON	0.194385	0.138543	0.291253	0.291218	-49.83	-110.20	0.139464	0.061535	28.258	55.58
AMEAN	0.133380	0.089306	0.174982	0.150952	-31.19	-69.03	0.088936	0.036044	**33.51**	**61.32**
GMEAN	0.128726	0.082123	0.138207	0.094319	N/A	N/A	0.085516	0.031132	**33.37**	**60.87**

FL_{max}, we use a representative synthetic traffic patterns with flit injection rate equals to 0.05 flits/cycle for ϵ (or RS_{max}), as depicted in Figure 5.8. we found that FL_{max} cannot exceed $2,400$ which can be counted by 12-bit counter. This injection rate is chosen based on the maximum possible traffic. In our experiments, we considered period P equal to one week for updating routing tables of each router and routing algorithm.

5.8.2 Results

For each benchmark, the aging-induced delay degradation of routers as well as the age imbalance (Δd) between routers are extracted for both 2D and 3D NoC in 3 years (9.3E+7 seconds) of execution. The following three schemes are compared:

- XY routing, which is not-adaptive and therefore non-aging aware routing (NAW). This method is intensified by our monitoring system using D-CAT to extract routers' ages online. For the 3D case, we used XYZ routing.
- Offline aging-aware routing through assigning budgets (OFAR), which is based on state-of-the-art works in [19, 20] and enhanced by our proposed online aging monitoring system for fairness. It assigns a lifetime budget for each router and defined as the fraction of the traffic that a stressed router should accept. Each routers' budget is predetermined using profiling for each benchmark.
- AROMa, our aging-aware adaptive routing and proposed online aging monitoring system.

Table 5.4 and Table 5.5 summarize the results for 2D NoC and 3D NoC, respectively. As shown in Table 5.4, AROMa improves the age of maximum-aged router by 33.5% on average compared to non-aging aware routing (NAW). For example, in RADIX the age of maximum

Table 5.5: Aging-induced delay degradation for maximum aged router and network age imbalance (Δ) in 3D NoC for 3 years of execution (9.3E+7 seconds).

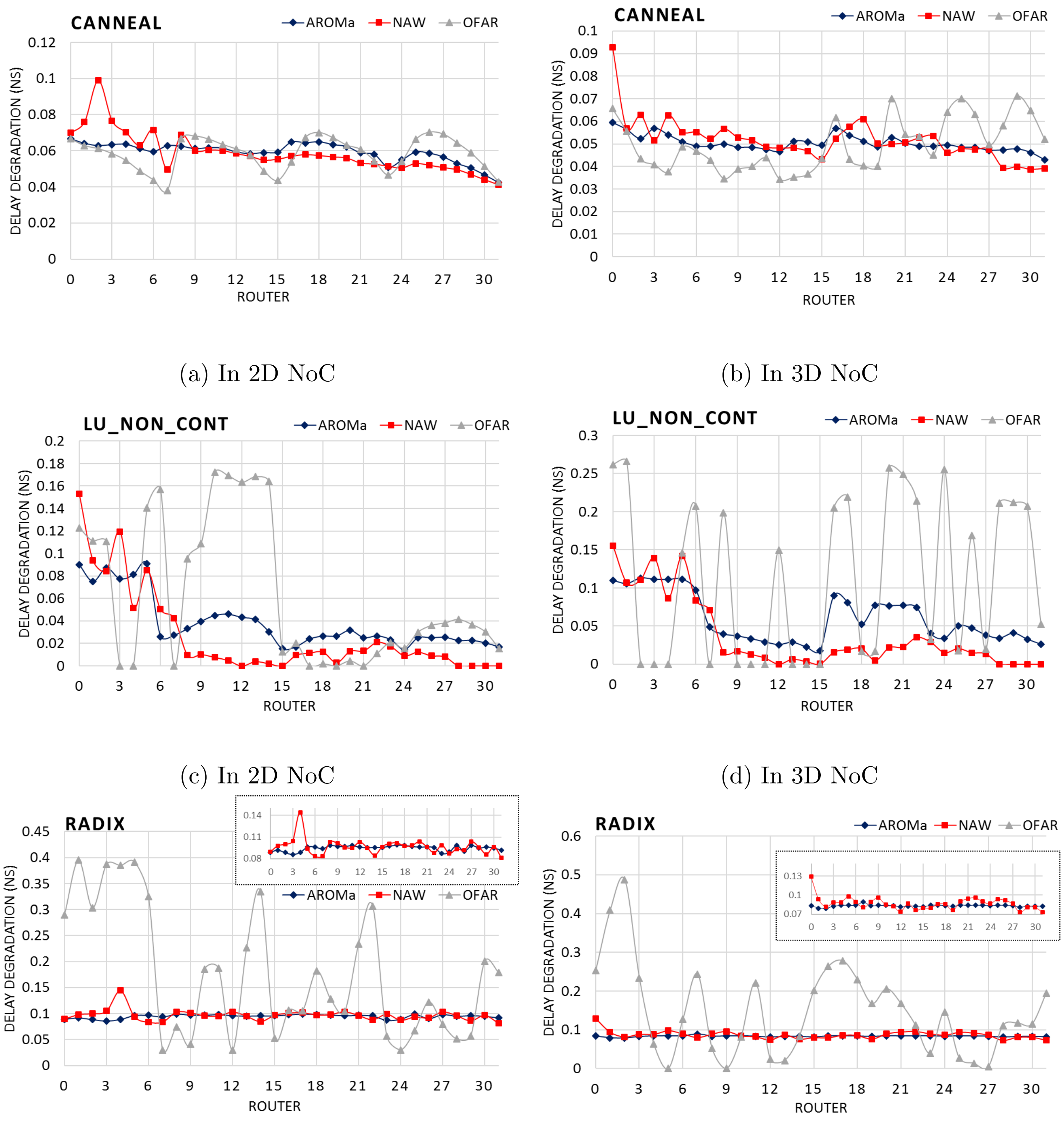
	NAW		OFAR				AROMa			
Benchmark	MAX(NS)	Δ(NS)	MAX(NS)	Δ(NS)	MAX(%)	Δ(%)	MAX(NS)	Δ(NS)	MAX(%)	Δ(%)
FFT	0.088736	0.049433	0.072167	0.038652	18.67	21.81	0.062064	0.018803	30.06	61.96
CANNEAL	0.092792	0.054076	0.071092	0.036751	23.39	32.04	0.059354	0.016520	36.04	69.45
LU_NON_CON	0.155106	0.155106	0.265872	0.265872	-71.41	-71.41	0.112998	0.095032	27.15	38.73
RADIX	0.128938	0.055948	0.486709	0.486709	-277.48	-769.94	0.088988	0.010294	30.98	81.60
SWAPTIONS	0.087378	0.047744	0.071672	0.037109	17.98	22.27	0.055814	0.011844	36.12	75.19
X264	0.097211	0.056439	0.077891	0.043849	19.87	22.31	0.062816	0.017808	35.38	68.45
BLACKSCHOLES	0.090148	0.051525	0.069848	0.035965	22.52	30.20	0.056309	0.013341	37.54	74.11
CHOLESKY	0.181058	0.137805	0.364778	0.364778	-101.47	-164.71	0.111719	0.017490	38.30	87.31
LU_CON	0.124743	0.088748	0.269684	0.269684	-116.19	-203.88	0.078841	0.009424	36.80	89.38
AMEAN	0.116234	0.077425	0.194413	0.175485	-51.57	-120.15	0.076545	0.023395	**34.26**	**71.80**
GMEAN	0.112332	0.069812	0.143291	0.100631	N/A	N/A	0.073733	0.017329	**34.05**	**70.02**

aged router in NAW scheme is 0.14487 ns while in our proposed method the maximum-aged router age is 0.098993 ns, which is equal to 31.43% improvement. Moreover, the age imbalance (Δ) between the maximum-aged and minimum-aged routers is improved by 61.31% in AROMa as compared to NAW. This shows how AROMa balances routers' ages effectively.

OFAR scheme assigns budgets to each router based on their load through profiling and then finds new source-destination shortest paths. OFAR is not able to balance age properly and reduce maximum-aged routers age. Our results in Table 5.4 also shows that maximum age and age imbalance (Δ) become worse by 31.19% and 69.03%, on average. For example, in X264 benchmark, the maximum age is improved by 29.67% and age imbalanced is improved by 30.23%. On the other hand, in LU_NON_CONT benchmark, there is no improvement and these values are -12.85% and -12.85%, respectively.

Similarly, Table 5.5 shows the results for 3D NoC. It is shown that on average the maximum age and age imbalance are improved by 34.26% and 71.80%, respectively. For example, in LU_CON benchmark the age of maximum aged router is 0.124743 ns in NAW while in AROMa it is 0.076545 ns, which means 36.80% improvement. In addition, the age imbalance (Δ) in NAW is 0.088748 ns and 0.009424 ns in AROMa, which is equal to a significant improvement of 89.38%.

In OFAR, both of 3D NoC and 2D NoC maximum age and age imbalance (Δ) are worsening by 31.19% and 69.03% as compared to NAW, respectively. Although the maximum age in FFT is improved by 26.71%, it is increased by 75.66% in CHOLESKY. Similarly, age imbalance is improved by 43.99% in CANNEAL while it is worsened by 110.20% in LU_CON benchmark. These results show that AROMa outperforms state-of-the-art works (OFAR) significantly. The main reason is that AROMa monitors age online and adaptively changes source-destination shortest paths to avoid maximum aged routers. In contrast, OFAR finds shortest paths and assigns age budgets to routers offline. Therefore, it cannot balance ages and reduce maximum ages properly. More will be detailed in the next subsection.



(a) In 2D NoC

(b) In 3D NoC

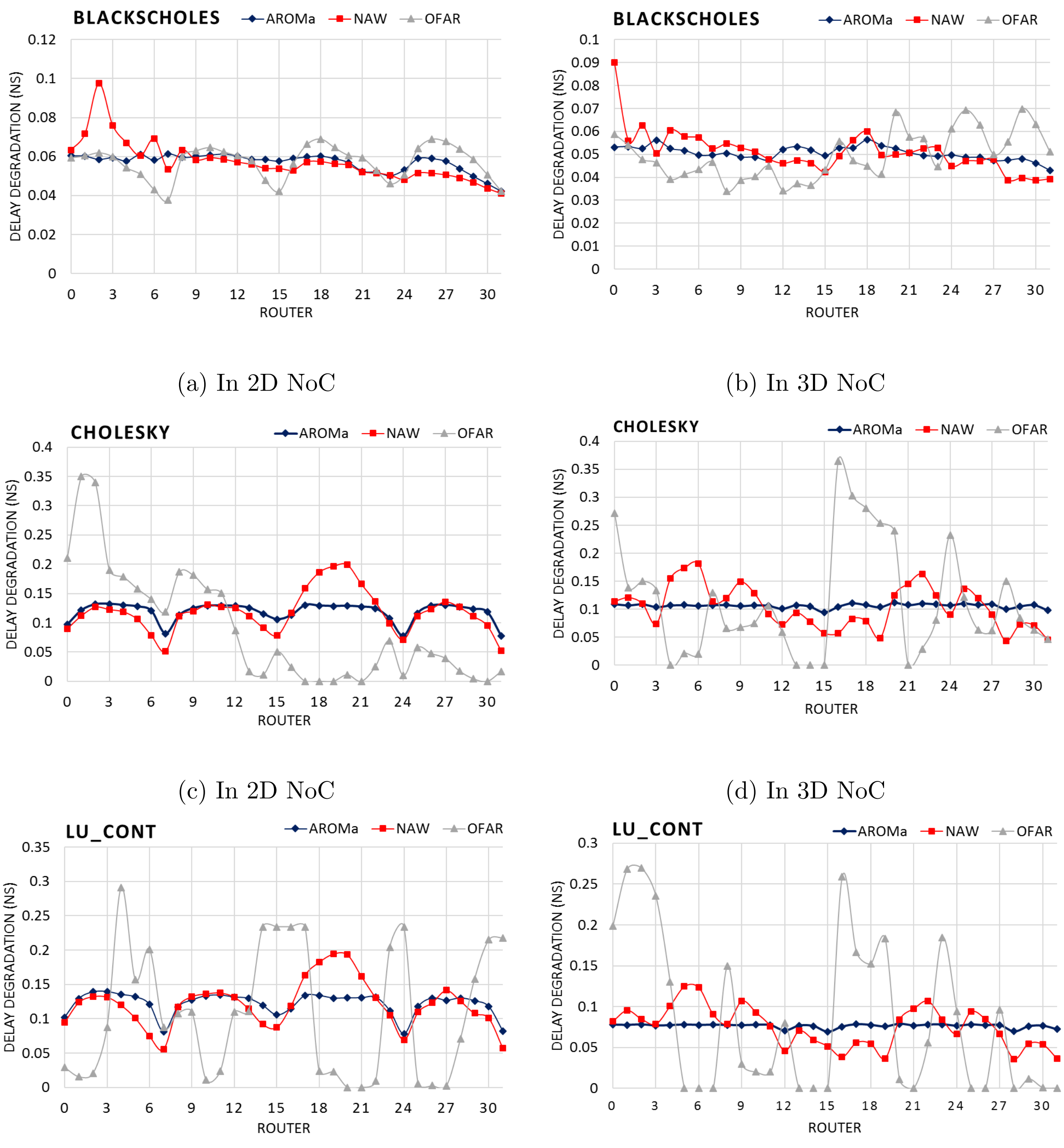
(c) In 2D NoC

(d) In 3D NoC

(e) In 2D NoC

(f) In 3D NoC

Figure 5.17: Age imbalance in 3 years for different routers in the network.



(a) In 2D NoC

(b) In 3D NoC

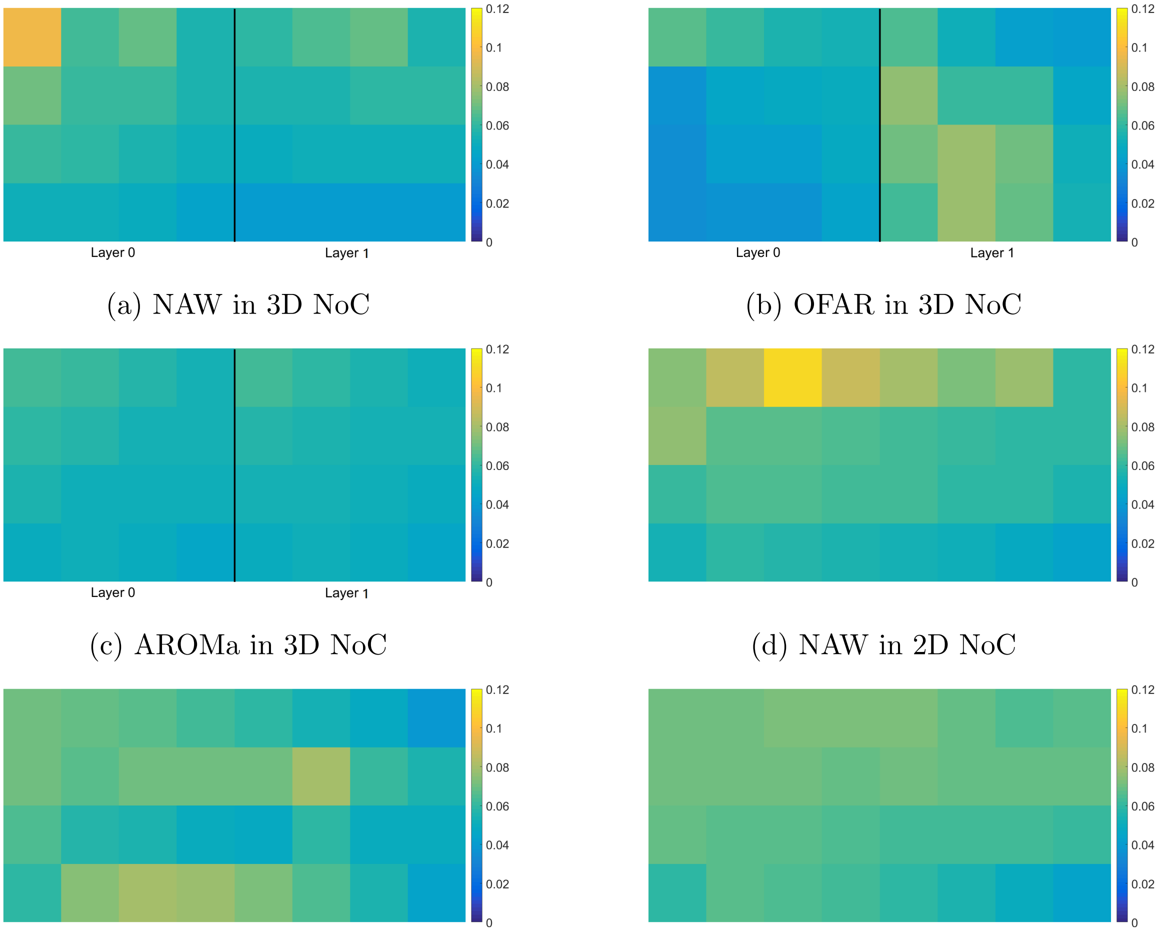
(c) In 2D NoC

(d) In 3D NoC

(e) In 2D NoC

(f) In 3D NoC

Figure 5.18: Age imbalance in 3 years for different routers in the network (continued).



(a) NAW in 3D NoC

(b) OFAR in 3D NoC

(c) AROMa in 3D NoC

(d) NAW in 2D NoC

(e) OFAR in 2D NoC

(f) AROMa in 2D NoC

Figure 5.19: Age imbalance between different routers in X264.

5.8.3 Analysis and Discussions

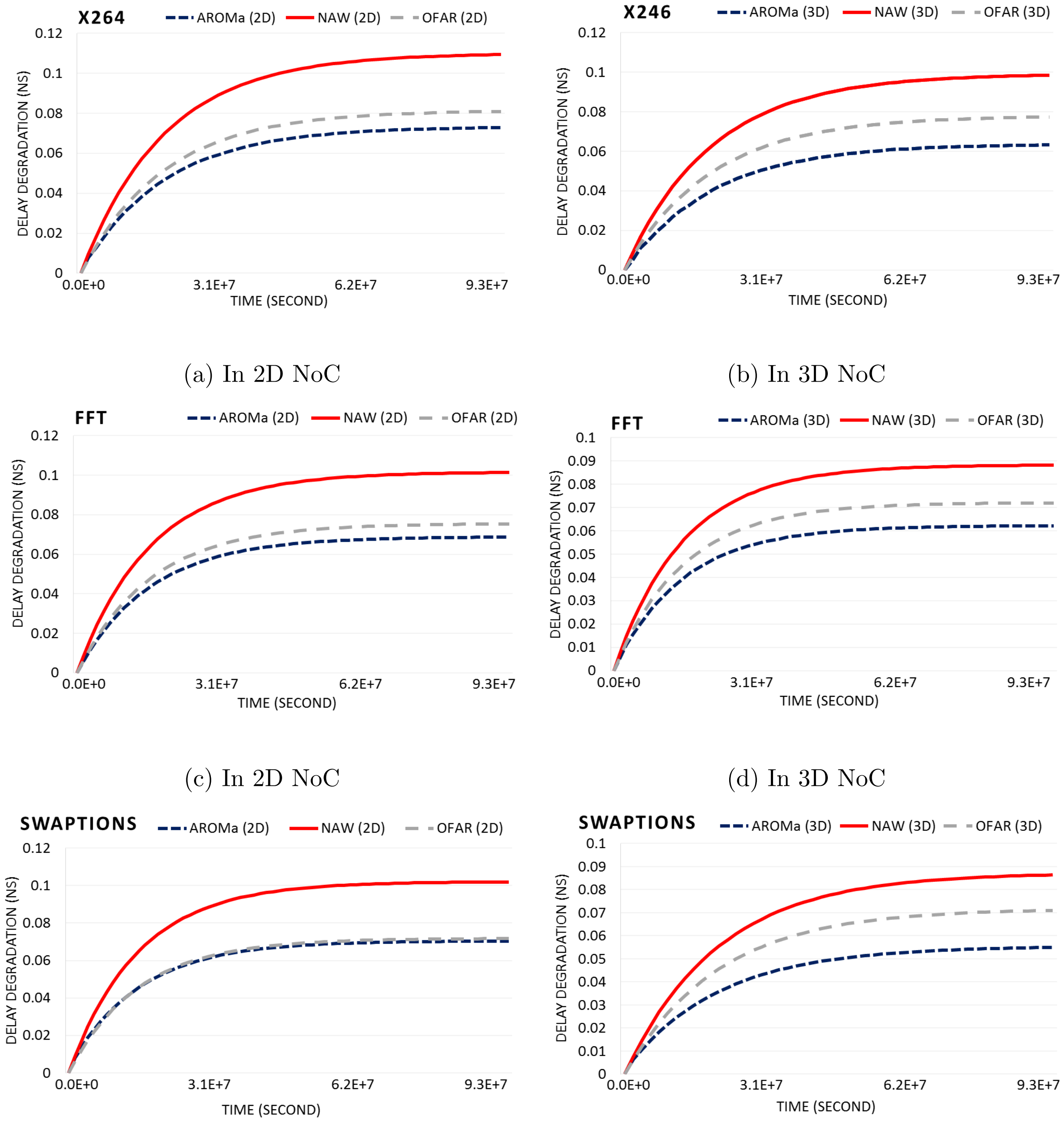
AROMa's main purpose is to balance ages between different routers in the network. It is done by avoiding highly aged routers and finding new shortest paths in each period P. This not only reduces the age of maximum aged router (through BTI recovery) but also decreases the difference between different router's ages, so-called age imbalance. Figure 5.17 and Figure 5.18, illustrate the age imbalance (Δ) for both 3D and 2D NoCs between all the routers in AROMa, NAW, and OFAR. The horizontal axis shows each router number in the network and the vertical axis shows the age (aging-induced delay degradation) of each router in the network. It can be seen that AROMa balance ages properly as compared to NAW and OFAR. For example, in RADIX on 3D NoC all routers ages are around 0.1 ns while in NAW the different routers are aged differently which needs to be improved. This becomes even worse in OFAR since the age of highly aged routers passed the guardband (0.16 ns). These routers are considered faulty and also pass their assigned offline budgets.

Furthermore, Figure 5.17 and Figure 5.18 show that in OFAR scheme some routers for certain benchmarks are overstressed because of offline budgeting. This will cause some of the routers to become faulty and pass the threshold of aging guardband (i.e. 16 ns). The reason is that in the offline profiling stage they were barely utilized, and are assigned higher budgets to diminish the utilization of highly overloaded routers. This unfair budgeting is not able to balance the age of routers to avoid aging other routers. For example, as shown in Figure 5.18.j, for CHOLESKY in 3D NoC, seven routers are highly stressed and become faulty. However, some of the routers are rarely used due to unfair budgeting. Nevertheless, we continued the simulation of the network after these routers become faulty, which leads to unpredictable network behavior.

The color map in Figure 5.19 illustrates the age imbalance of routers in X264, as an example for both 2D and 3D NoC using NAW, OFAR, and AROMa for 3-year execution. Each square represents a router's age. The lighter color indicates higher age. It can be seen that the bottom layer (layer 0) in NAW is aged more than the upper layer (layer 1) (Figure 5.19.a). Therefore, OFAR assigns more budget to layer 1 routers to reduce the age in layer 0 but increases the age in layer 1 routers (Figure 5.19.c). As shown in Figure 5.19.c, by swapping between different shortest-paths, AROMa balances ages among both layer 0 and layer 1, uniformly. In contrast, routers ages in 2D NoC for both NAW and OFAR schemes are not uniformly distributed (Figure 5.19.d and Figure 5.19.e), while AROMa distributes ages between routers in the network properly.

Furthermore, it can be observed from Figure 5.17, Figure 5.18 and Figure 5.19 that routers age more in 2D NoC than 3D NoC even though the temperature is higher in 3D NoC upper layers. The main reason is that in 2D NoC the shortest paths between each source and destination are usually longer. This means that a flit requires more time to traverse the NoC through more routers. This results in more usage of routers and subsequently more delay degradation for them. All in all, it can be concluded that 3D NoC architecture performs better in term of aging as compared to 2D NoC.

Figure 5.20 illustrates the trend of maximum aged router's delay degradation in 3 years (9.3E+7 seconds) of execution in both 3D and 2D NoCs for three selected benchmarks. The maximum age after 3 years of execution for 2D NoC is higher than 3D NoC. For example, the maximum age of routers in 2D NoC is 12.89%, 11.24%, and 14.28% higher than 3D NoC for FFT, X264, and SWAPTIONS, respectively. Figure 5.20.c and Figure 5.20.d show the delay degradation trend for FFT. As depicted in the figure AROMa outperforms OFAR by 14% and 8.9% in 3D and 2D NoC, respectively. Based on the holistic results in Table 5.4, Table 5.5, Figure 5.17, Figure 5.18, and Figure 5.20, we conclude that AROMa performs better in 3D NoC and decreases maximum age even more. In contrary, on average OFAR shows better performance in 2D as compared to 3D in term of lowering the maximum age.



(a) In 2D NoC

(b) In 3D NoC

(c) In 2D NoC

(d) In 3D NoC

(e) In 2D NoC

(f) In 3D NoC

Figure 5.20: Delay degradation for 3 years (9.3E+7 seconds) for maximum aged routers in the network.

5.8.4 Overhead

Area

As shown in Figure 5.9, for online monitoring system we need to add two parallel counters (12-bit and 14-bit), seven 14-bit subtractors (Maximum number of ports in a 3D NoC is seven), seven 4-bit MUXes, and seven AND logic gates. We embedded them into our router architecture from [1] and used Cadence tools to extract area overhead. Our analysis shows that the area overhead is negligible ($\sim 0.55\%$). The area overhead for previous methods, that OFAR deployed them, in [20] and [131] is ($\sim 4\%$) and ($\sim 1.2\%$), respectively. One can conclude that our online monitoring system imposes a negligible area overhead to each router as compared to other reported works.

As shown in Figure 5.9 and Figure 5.10, each D-CAT has three columns of information. The first one corresponds to *residence-time* (rs) which can be represented by two bytes. Similarly, the second column represents *number-of-flits* (fl), which also can be represented by two bytes. In addition, to store the aging-induced delay degradation corresponding to each fl and rs pair four bytes in the third column is used. In our experiments, the number of steps is 482 (in Algorithm 6), which means our D-CATs have 482 entries considering that each step is 50 flits. We observed that a change in the number of flits by 50 does not change the power consumption and temperature noticeably. Therefore, the total amount of memory that is required for each D-CAT is $482 \times (2 + 2 + 4)$ bytes which is equal to ($\sim 3.8KB$).

Energy-Delay-Product-Per-Flit

In Figure 5.21, we illustrate the *Energy-Delay-Product-Per-Flit* (EDDPF) of each benchmark for AROMa and OFAR schemes compared to NAW in both 2D and 3D NoCs. The EDDPF on average is 1.53%, 1.88%, 1.67%, and 6.91% for AROMa in 2D NoC, AROMa in 3D,

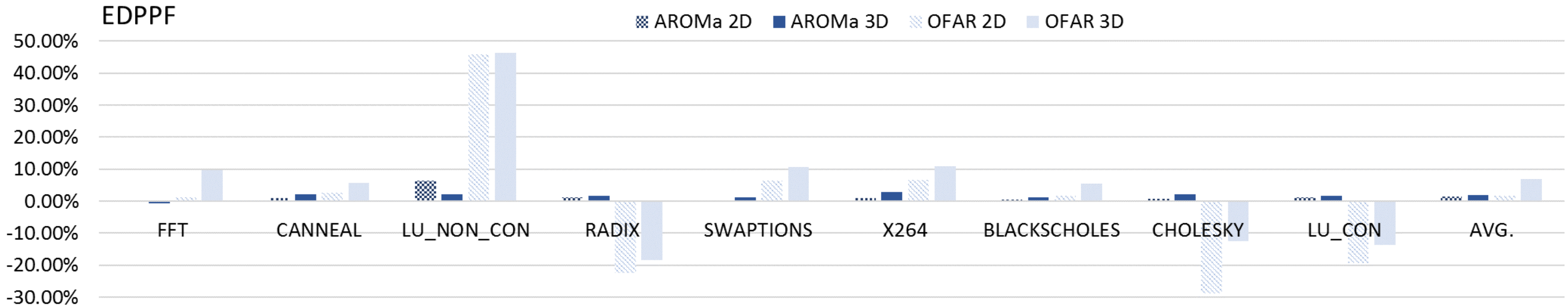


Figure 5.21: EDPPF overhead for 3-year (9.3E+7 seconds) execution time in each benchmark for different schemes.

OFAR in 2D, and OFAR in 3D, respectively. It can be concluded that OFAR has higher EDDPF overhead as compared to AROMa for both 2D and 3D NoCs on average. However, the EDPPF is improved for some of the benchmarks in OFAR scheme. In OFAR, there are faulty routers since they are overstressed (i.e. overaged) because of offline budgeting. As we mentioned in Subsection 5.8.3, the overstressed routers caused by unfair budgeting in OFAR make network behavior unpredictable. For example, in RADIX, the EDDPF overhead is improved while in LU_NON_CON it is increased.

In addition, EDPPF in 2D NoC is performing better than 3D NoC for all benchmarks expect LU_NON_CON in AROMa. Even though 3D NoC delay is decreased, the EDDPF increases as compared to 2D because of the increase in power and energy consumption.

Network latency

Figure 5.22 illustrates latency comparison for both 3D and 2D NoCs using different techniques for uniform random traffic distribution. It shows that latency in 3D NoC saturates later than 2D NoCs as expected. In both cases, OFAR saturates with lower injection rates than both AROMa and NAW. The main reason is that the shortest-path selection in OFAR is not aware of online changes of the network that may lead to over-utilization of some parts of the network, resulting in additional latency. As illustrated, AROMa for both 2D and 3D NoCs has negligible latency to the network as compared to NAW and saturates in slightly lower injection rates.

Link utilization

Figure 5.23 demonstrates link utilization of AROMa and OFAR schemes for 2D and 3D NoC. For our set of benchmarks, the average link utilization is 0.7%, 0.53%, 2.32%, and 14.69% for AROMa in 2D NoC, AROMa in 3D NoC, OFAR in 2D NoC, and OFAR in 3D

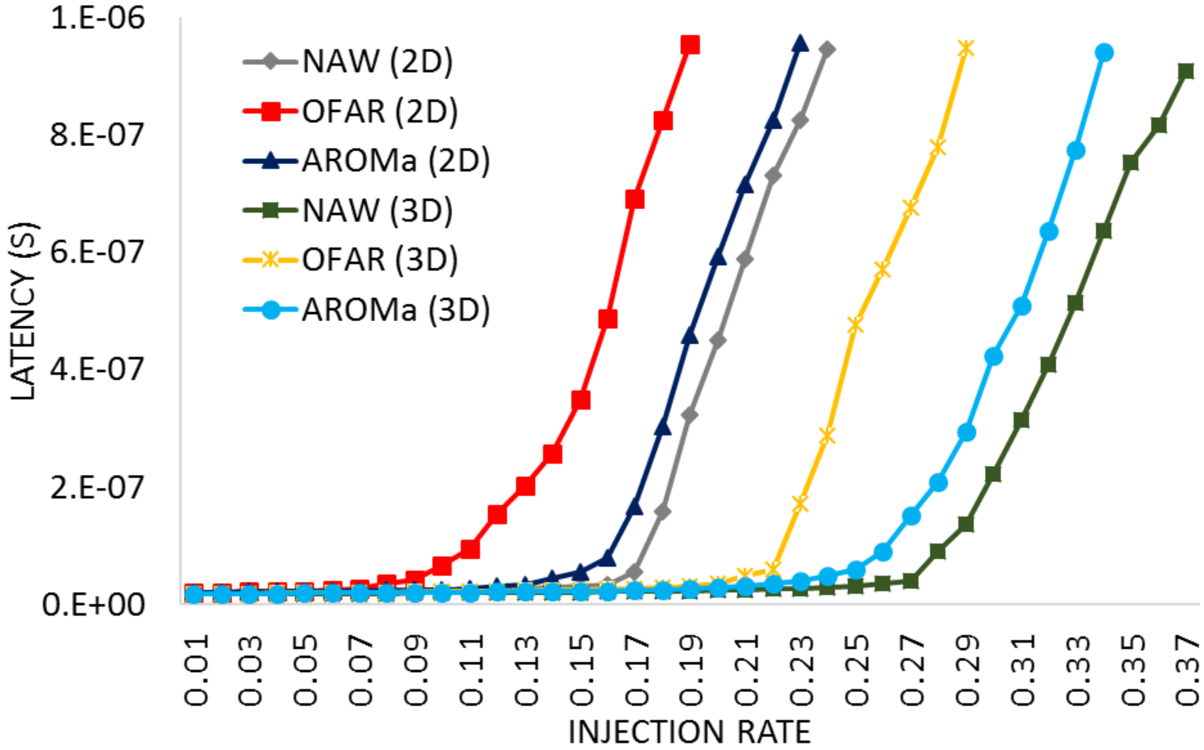


Figure 5.22: Latency in different injection rates.

NoC, respectively. Except for LU_NON_CON benchmark OFAR in 3D NoC has higher link utilization as compared to AROMa. Similarly, 2D NoC OFAR has higher link utilization in comparison to AROMa except in RADIX benchmark. Link utilization depends on benchmarks' behaviors as well as the routing algorithm. OFAR uses certain paths to traverse flits to avoid highly utilized routers that are determined during the offline network evaluation. This leads to lower utilization of connected links to such routers but migrates loads to another set of links connected to low utilized routers. Therefore, OFAR chooses longer paths to traverse flits which results in more unnecessary misleading higher link utilization (Figure 5.23) in comparison to AROMa and NAW for the same traffic. In all, OFAR is not able to fairly utilize resources in the NoC and transferring the load (age or problem) from one region to another. As our results show, AROMa outperforms OFAR in term of fair utilization of network resources which leads to a better balance of ages.

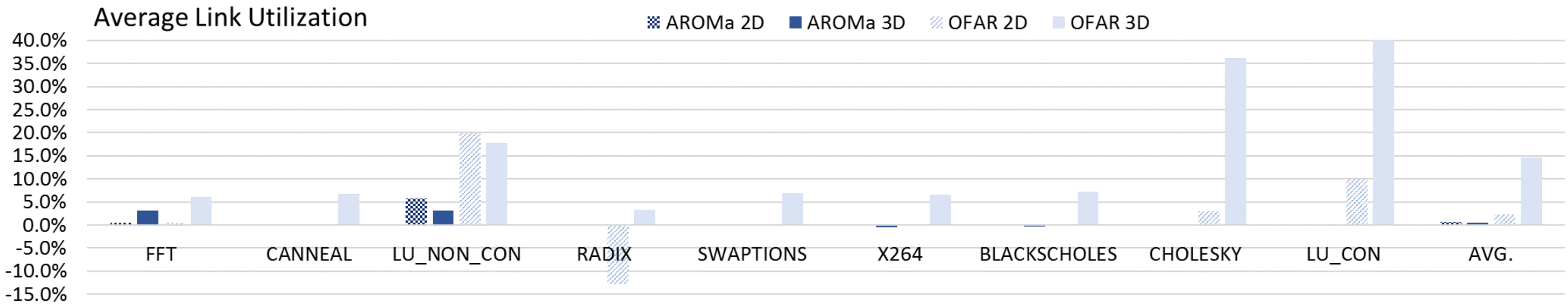


Figure 5.23: Link utilization overhead for 3-year (9.3E+7 seconds) execution time in each benchmark for different schemes.

5.9 Conclusion

In this chapter, we proposed AROMa, an adaptive deadlock-free aging-aware routing algorithm along with an online aging monitoring system for 3D NoCs. Temperature is a fundamental challenge in 3D NoCs which can significantly change both BTI and HCI aging mechanisms. In addition, aging-induced delay degradation is a function of stress quantified by usage. We introduce D-CAT which stores delay degradation for each temperature and stress pairs. Our online monitoring system utilizes D-CAT for each layer of the 3D NoC to keep track of each router's age. Moreover, AROMa finds different shortest paths between each source-destination pair to avoid highly aged router at each period of time. Therefore, highly-aged routers potentially can recover from BTI-induced delay degradation because of our adaptive and aging-aware routing. Our extensive experimental evaluation for SPLASH-2 and PARSEC benchmarks using gem5 in full system mode for both 2D and 3D NoC shows that AROMa outperforms state-of-the-art work (OFAR). AROMa is significantly better than OFAR not only in terms of age imbalance between different routers but also minimizing maximum aged router age. Furthermore, our results show that 3D NoC is more resistant against aging as compared to 2D NoC even though the temperature may go higher in the upper layers. The main reason is that in 3D NoC paths are shorter and router's usage (stress) decreases. AROMa improves age imbalance by 60% and 72% in 2D and 3D NoC, respectively, in comparison to non-aging aware techniques. In addition, the maximum age of routers decreases by 33.51% and 34.26% in 2D and 3D NoC, respectively.

Chapter 6

Conclusion and Future Work

In this chapter, we give a summary of our achievements throughout this research, followed by a set of future direction research topics that potentially can improve the contributions of this dissertation.

6.1 Conclusion Remarks

In this dissertation, we demonstrate a simulation flow that combines the system-level simulation and thermal simulation tools to provide an accurate thermal analysis of 3D ICs. We also propose a TTSVs placement method to assign TTSV blocks along lateral and vertical directions of the core layers and close to the hot spot regions, where TTSVs can further penetrate cache layers to facilitate heat dissipation. Simulation flow and TTSVs placement method are performed on the x86 based Nehalem floorplan; results show that 3D Nehalem has considerable advantages in footprint, performance and power consumption. After adding TTSVs blocks, the peak chip temperature is reduced by 5%-25% with an area overhead of 6%. Higher frequency also increases the hot spot temperature significantly for 3D ICs, a larger TTSVs area is expected for high frequency applications. We also present a hierarchical floorplanning approach for a 3D Nehalem-based multicore processor, which optimizes the peak temperature, wirelength and area of the floorplan through a SA-based TTSV placement optimization algorithm. Our simulation results show that optimally arranged TTSVs can effectively reduce the peak temperature with moderate sacrifices in wirelength and area overhead. The peak temperature decreases consistently with the TTSV area overhead while the wirelength change is strongly related to the TTSV placement, which is uniquely optimized with different TTSV area overheads. Moreover, a critical benchmark can be selected for guiding the floorplan optimization, and the optimized floorplan is applicable to other Splash-2 benchmarks without further modification.

In addition, we proposed an online monitoring technique for aging in NoC routers, which is utilized in our aging-aware routing algorithm. The router is analyzed for different number of flits associated with different temperature and stress to extract a CAT. CAT is placed in one of the middle cores that has minimal distance to the other cores inside the network, which can be accessed by each NoC router based on their number of flits and resident time during a given period of time. Our experimental analysis shows 39% and 52% improvement on critical path degradation of maximum aged router and aging imbalance, respectively, with negligible overheads. Furthermore, we proposed AROMa, an adaptive deadlock-free aging-aware routing algorithm along with an online aging monitoring system for 3D NoCs. We also introduced D-CAT which stores delay degradation for each temperature and stress pairs. Our online monitoring system utilizes D-CAT for each layer of the 3D NoC to keep track of each router's age. Moreover, AROMa finds different shortest paths between each source-destination pair to avoid highly aged router at each period of time. Therefore, highly-aged routers may get a chance to recover from BTI-induced delay degradation because of our adaptive and aging-aware routing. Our extensive experimental evaluation for SPLASH-2 and PARSEC benchmarks using gem5 in full system mode for both 2D and 3D NoC shows that AROMa outperforms state-of-the-art work (OFAR). AROMa is significantly better than

OFAR not only in terms of age imbalance between different routers but also minimizing maximum aged router age. Furthermore, our results show that 3D NoC is more resistant against aging as compared to 2D NoC even though the temperature may go higher in the upper layers. The main reason is that in 3D NoC paths are shorter and router's usage (stress) decreases. AROMa improves age imbalance by 60% and 72% in 2D and 3D NoC, respectively, in comparison to non-aging aware techniques. In addition, the maximum age of routers decreases by 33.51% and 34.26% in 2D and 3D NoC, respectively.

6.2 Future Directions

This dissertation prompted several ideas which are achievable as future directions. The following subsections briefly identify them.

6.2.1 Machine Learning method for THermal Aware Floorplanning (ML-THAF)

One direction for future study is to incorporate two machine learning techniques in order to increase the search space many folds using deep reinforcement learning, which will replace the described simulated annealing algorithm. This will speed up the evaluation of the floorplaning via a trained predictive model using supervised learning by using many kernel *Convolutional Neural Network* (CNN) approach that investigates several extracted features. The first technique could use a predictive model rather than a full simulation evaluation. Each iteration step of the search evaluation is enhanced because of the reduced execution time as compared to a full simulation. A full simulation will be applied initially to create a diverse data set used in the supervised training. ML-THAF predictive model and its deep reinforcement learning are demonstrated in the following subsections.

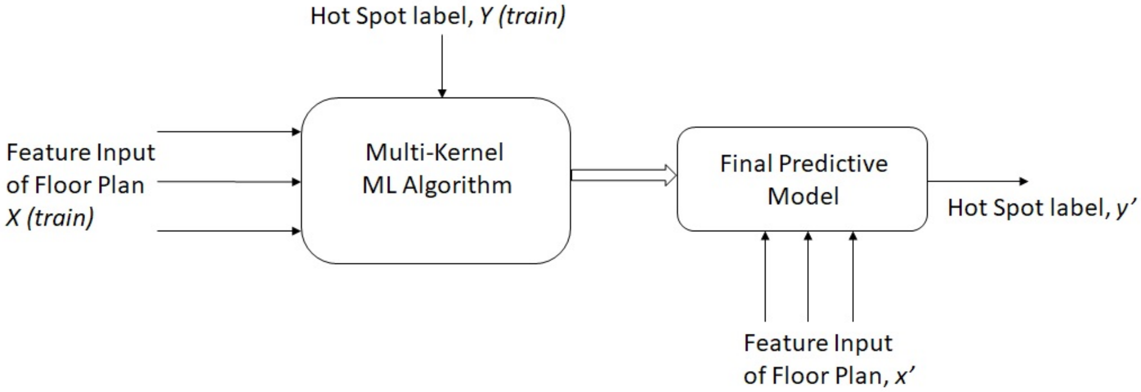

Hot Spot label, *Y (train)*
Feature Input of Floor Plan *X (train)*
Multi-Kernel ML Algorithm
Final Predictive Model
Hot Spot label, *y'*
Feature Input of Floor Plan, *x'*

Figure 6.1: ML-THAF predictive model.

Predictive Model

Initially, supervised training must take place, a set of balanced data points of positive and negative floorplans with proper labels are used in the training. The final predictive model will be evaluated on its performance using evaluation set that is generated from the simulation. Once the resulting accuracy is high enough below a specific allowed error threshold, the predictive model is ready to be employed in the reinforcement learning cycle. Figure 6.1 shows the predictive model, first the floorplan feature is extracted for floorplan (X) and is matched against their expected output (Y). This step is possible because the data set is already being identified as a pair of input and output (X, Y). Hence, the final predictive model can be generated using multi-kernel machine learning algorithm that is composed of complex neural network trained by feedforward then feedback propagation learning method. When the predictive model is ready, a new (X') evaluation can be generated using the predictive model denoted by (Y').

Deep Reinforcement Learning

Deep Reinforcement Learning (Deep-RL) then is applied, which has been quite successful in solving NP-hard combinatorial optimization problems. Figure 6.2 is demonstrating a

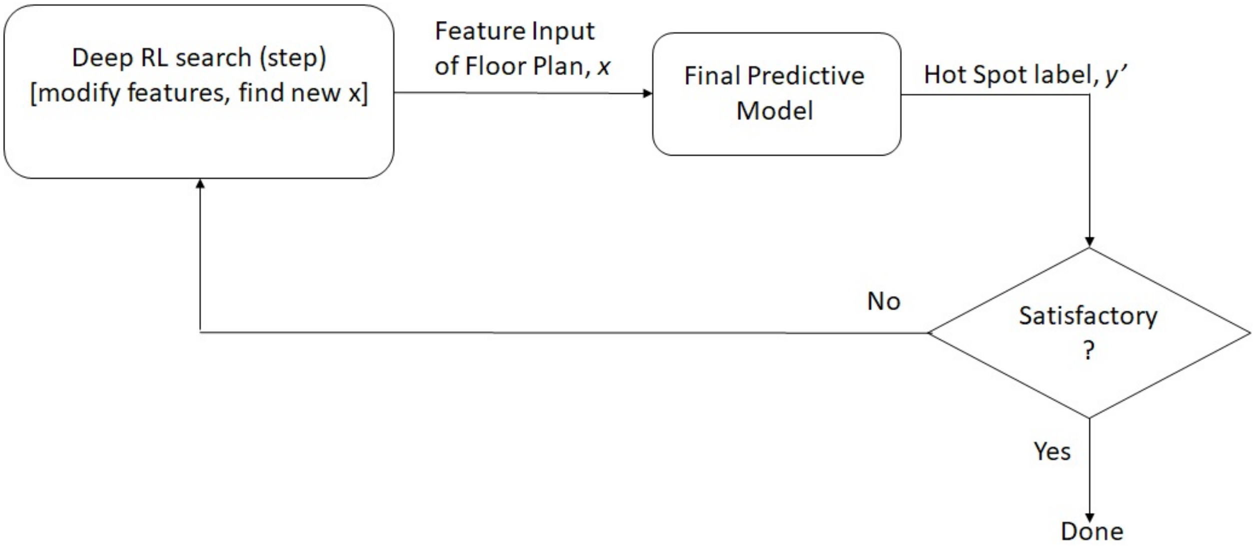


Figure 6.2: ML-THAF deep reinforcement learning.

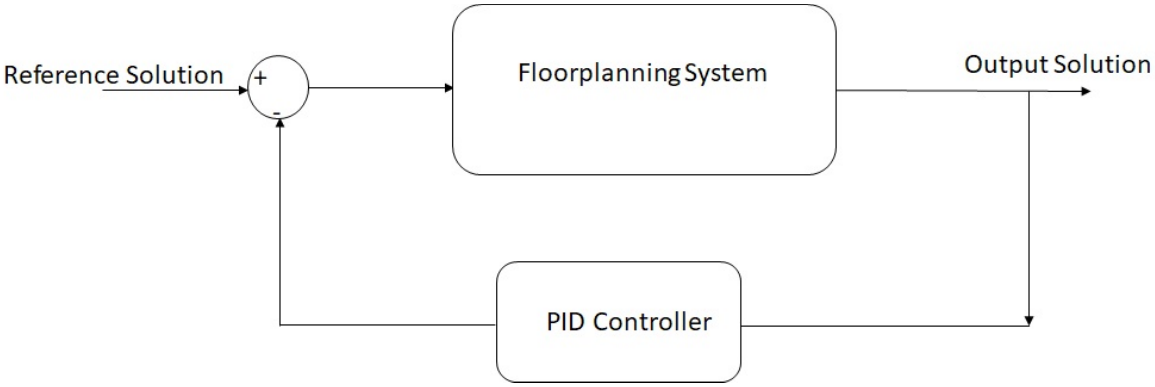


Figure 6.3: Control system with PID controller.

potential general framework. Recurrent Neural Networks is used as the model for the Deep-RL, the parameters are modified at each iteration of the search to produce new feature inputs, which will be evaluated by the predictive model.

6.2.2 Utilizing PID Controller

Another future direction is to take advantage of control system theory in directing the search for a suboptimal floorplan solution with the inclusion of TTSVs. *Proportional, Integral, and Derivative* (PID) controllers can be employed as depicted in Figure 6.3.

6.2.3 AROMA Improvement

In AROMa, routers' temperatures are estimated by their *number-of-flits (fl)* and corresponding *residence-times (rs)*, the impact of their surrounding circuits' temperatures are not considered for this study. The required analysis can be investigated as future work. In addition, process variation inclusion in our model would enhance its accuracy.

Bibliography

[1] Open source noc router rtl, http://nocs.stanford.edu/cgi-bin/trac.cgi/wiki/resources/router.

[2] International technology roadmap for semiconductors 2.0, edition 2015, http://www.semiconductors.org. 2015.

[3] N. Agarwal, T. Krishna, L.-S. Peh, and N. K. Jha. Garnet: A detailed on-chip network model inside a full-system simulator. In *Performance Analysis of Systems and Software, 2009. ISPASS 2009. IEEE International Symposium on*, pages 33–42. IEEE, 2009.

[4] A. Agrawal, J. Torrellas, and S. Idgunji. Xylem: enhancing vertical thermal conduction in 3d processor-memory stacks. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 546–559. ACM, 2017.

[5] M. O. Agyeman, A. Ahmadinia, and N. Bagherzadeh. Performance and energy aware inhomogeneous 3d networks-on-chip architecture generation. *IEEE Transactions on Parallel and Distributed Systems*, 27(6):1756–1769, 2016.

[6] M. A. Ahmed and M. Chrzanowska-Jeske. Tsv capacitance aware 3-d floorplanning. In *2013 IEEE International 3D Systems Integration Conference (3DIC)*, pages 1–6. IEEE, 2013.

[7] 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. IEEE, 2012.

[8] H. Aljazzar and S. Leue. K: A heuristic search algorithm for finding the k shortest paths. *Artificial Intelligence*, 175(18):2129–2154, 2011.

[9] A. Alqahtani, Z. Ren, J. Lee, and N. Bagherzadeh. System-level analysis of 3d ics with thermal tsvs. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 14(3):37, 2018.

[10] D. M. Ancajas, K. Bhardwaj, K. Chakraborty, and S. Roy. Wearout resilience in nocs through an aging aware adaptive routing algorithm. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(2):369–373, 2015.

[11] D. M. Ancajas, K. Chakraborty, and S. Roy. Proactive aging management in heterogeneous nocs through a criticality-driven routing approach. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 1032–1037. EDA Consortium, 2013.

[12] D. M. Ancajas, J. M. Nickerson, K. Chakraborty, and S. Roy. Hci-tolerant noc router microarchitecture. In *Design Automation Conference (DAC), 2013 50th ACM/EDAC/IEEE*, pages 1–10. IEEE, 2013.

[13] K. Arabi, K. Samadi, and Y. Du. 3d vlsi: a scalable integration beyond 2d. In *Proceedings of the 2015 Symposium on International Symposium on Physical Design*, pages 1–7. ACM, 2015.

[14] J. H. Bahn and N. Bagherzadeh. A generic traffic model for on-chip interconnection networks. *Network on Chip Architectures*, page 22, 2008.

[15] A. Bar-Cohen and P. Wang. Thermal management of on-chip hot spot. *Journal of Heat Transfer*, 134(5):051017, 2012.

[16] K. Bernstein, P. Andry, J. Cann, P. Emma, D. Greenberg, W. Haensch, M. Ignatowski, S. Koester, J. Magerlein, R. Puri, et al. Interconnects in the third dimension: Design challenges for 3d ics. In *2007 44th ACM/IEEE Design Automation Conference*, pages 562–567. IEEE, 2007.

[17] K. Bernstein, D. J. Frank, A. E. Gattiker, W. Haensch, B. L. Ji, S. R. Nassif, E. J. Nowak, D. J. Pearson, and N. J. Rohrer. High-performance cmos variability in the 65-nm regime and beyond. *IBM journal of research and development*, 50(4.5):433–449, 2006.

[18] E. Beyne, P. De Moor, W. Ruythooren, R. Labie, A. Jourdain, H. Tilmans, D. S. Tezcan, P. Soussan, B. Swinnen, and R. Cartuyvels. Through-silicon via and die stacking technologies for microsystems-integration. In *Electron Devices Meeting, 2008. IEDM 2008. IEEE International*, pages 1–4. IEEE, 2008.

[19] K. Bhardwaj, K. Chakraborty, and S. Roy. An milp-based aging-aware routing algorithm for nocs. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012*, pages 326–331. IEEE, 2012.

[20] K. Bhardwaj, K. Chakraborty, and S. Roy. Towards graceful aging degradation in nocs through an adaptive routing algorithm. In *Proceedings of the 49th Annual Design Automation Conference*, pages 382–391. ACM, 2012.

[21] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula. Predictive modeling of the nbti effect for reliable design. In *Custom Integrated Circuits Conference, 2006. CICC'06. IEEE*, pages 189–192. IEEE, 2006.

[22] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, et al. The gem5 simulator. *ACM SIGARCH Computer Architecture News*, 39(2):1–7, 2011.

[23] T. Bjerregaard and S. Mahadevan. A survey of research and practices of network-on-chip. *ACM Computing Surveys (CSUR)*, 38(1):1, 2006.

[24] B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. H. Loh, D. McCaule, P. Morrow, D. W. Nelson, D. Pantuso, et al. Die stacking (3d) microarchitecture. In *Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 469–479. IEEE Computer Society, 2006.

[25] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny. Qnoc: Qos architecture and design process for network on chip. *Journal of systems architecture*, 50(2):105–128, 2004.

[26] P. Budhathoki, A. Henschel, and I. A. M. Elfadel. Thermal-driven 3d floorplanning using localized tsv placement. In *IC Design & Technology (ICICDT), 2014 IEEE International Conference on*, pages 1–4. IEEE, 2014.

[27] J. Burns. Tsv-based 3d integration. In *Three Dimensional System Integration*, pages 13–32. Springer, 2011.

[28] T. E. Carlson, W. Heirman, and L. Eeckhout. Sniper: exploring the level of abstraction for scalable and accurate parallel multi-core simulation. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, page 52. ACM, 2011.

[29] C.-P. Chen, Y. Weng, and G. Subbarayan. Topology optimization for efficient heat removal in 3d packages. In *Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm), 2016 15th IEEE Intersociety Conference on*, pages 238–244. IEEE, 2016.

[30] H.-T. Chen, H.-L. Lin, Z.-C. Wang, and T. Hwang. A new architecture for power network in 3d ic. In *2011 Design, Automation & Test in Europe*, pages 1–6. IEEE, 2011.

[31] K.-C. J. Chen, C.-H. Chao, and A.-Y. A. Wu. Thermal-aware 3d network-on-chip (3d noc) designs: routing algorithms and thermal managements. *IEEE Circuits and Systems Magazine*, 15(4):45–69, 2015.

[32] K.-N. Chen and C. S. Tan. Integration schemes and enabling technologies for three-dimensional integrated circuits. *IET Computers & Digital Techniques*, 5(3):160–168, 2011.

[33] T.-Y. Chiang, S. J. Souri, C. O. Chui, and K. C. Saraswat. Thermal analysis of heterogeneous 3d ics with various integration scenarios. In *Electron Devices Meeting, 2001. IEDM'01. Technical Digest. International*, pages 31–2. IEEE, 2001.

[34] H.-C. Chien, J. H. Lau, Y.-L. Chao, M.-J. Dai, R.-M. Tain, L. Li, P. Su, J. Xue, and M. Brillhart. Thermal evaluation and analyses of 3d ic integration sip with tsvs for network system applications. In *Electronic Components and Technology Conference (ECTC), 2012 IEEE 62nd*, pages 1866–1873. IEEE, 2012.

[35] M. Cho, S. T. Kim, C. Tokunaga, C. Augustine, J. P. Kulkarni, K. Ravichandran, J. W. Tschanz, M. M. Khellah, and V. De. Postsilicon voltage guard-band reduction in a 22 nm graphics execution core using adaptive voltage scaling and dynamic power gating. *IEEE Journal of Solid-State Circuits*, 52(1):50–63, 2017.

[36] E. Colgan, P. Andry, B. Dang, J. Magerlein, J. Maria, R. Polastre, and J. Wakil. Measurement of microbump thermal resistance in 3d chip stacks. In *Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM), 2012 28th Annual IEEE*, pages 1–7. IEEE, 2012.

[37] E. Colgan, R. Polastre, J. Knickerbocker, J. Wakil, J. Gambino, and K. Tallman. Measurement of back end of line thermal resistance for 3d chip stacks. In *Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM), 2013 29th Annual IEEE*, pages 23–28. IEEE, 2013.

[38] J. Cong and Y. Zhang. Thermal via planning for 3-d ics. In *Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*, pages 745–752. IEEE Computer Society, 2005.

[39] T. H. Cormen. *Introduction to algorithms*. MIT press, 2009.

[40] A. K. Coskun, J. L. Ayala, D. Atienza, T. S. Rosing, and Y. Leblebici. Dynamic thermal management in 3d multicore architectures. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 1410–1415. European Design and Automation Association, 2009.

[41] W. J. Dally and C. L. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. 1988.

[42] B. Dang, M. S. Bakir, D. C. Sekar, C. R. King Jr, and J. D. Meindl. Integrated microfluidic cooling and interconnects for 2d and 3d chips. *IEEE Transactions on Advanced Packaging*, 33(1):79–87, 2010.

[43] A. Das, A. Kumar, and B. Veeravalli. Reliability and energy-aware mapping and scheduling of multimedia applications on multiprocessor systems. *IEEE Transactions on Parallel and Distributed Systems*, 27(3):869–884, 2016.

[44] W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon. Demystifying 3d ics: The pros and cons of going vertical. *IEEE Design & Test of Computers*, 22(6):498–510, 2005.

[45] M. Ebrahimi and M. Daneshtalab. Ebda: A new theory on design and verification of deadlock-free interconnection networks. In *ISCA*, pages 1–13, 2017.

[46] M. Ebrahimi, M. Daneshtalab, F. Farahnakian, J. Plosila, P. Liljeberg, M. Palesi, and H. Tenhunen. Haraq: congestion-aware learning model for highly adaptive routing algorithm in on-chip networks. In *Networks on Chip (NoCS), 2012 Sixth IEEE/ACM International Symposium on*, pages 19–26. IEEE, 2012.

[47] D. Edelstein, H. Rathore, C. Davis, L. Clevenger, A. Cowley, T. Nogami, B. Agarwala, S. Arai, A. Carbone, K. Chanda, et al. Comprehensive reliability evaluation of a 90 nm cmos technology with cu/pecvd low-k beol. In *Reliability Physics Symposium Proceedings, 2004. 42nd Annual. 2004 IEEE International*, pages 316–319. IEEE, 2004.

[48] N. Eisley, L.-S. Peh, and L. Shang. In-network cache coherence. In *Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 321–332. IEEE Computer Society, 2006.

[49] P. Emma, A. Buyuktosunoglu, M. Healy, K. Kailas, V. Puente, R. Yu, A. Hartstein, P. Bose, and J. Moreno. 3d stacking of high-performance processors. In *High Performance Computer Architecture (HPCA), 2014 IEEE 20th International Symposium on*, pages 500–511. IEEE, 2014.

[50] P. G. Emma and E. Kursun. Is 3d chip technology the next growth engine for performance improvement? *IBM journal of research and development*, 52(6):541–552, 2008.

[51] D. Eppstein. Finding the k shortest paths. *SIAM Journal on computing*, 28(2):652–673, 1998.

[52] B. S. Feero and P. P. Pande. Networks-on-chip in a three-dimensional environment: A performance evaluation. *IEEE Transactions on computers*, 58(1):32–45, 2009.

[53] D. Fick, A. DeOrio, G. Chen, V. Bertacco, D. Sylvester, and D. Blaauw. A highly resilient routing algorithm for fault-tolerant nocs. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 21–26. European Design and Automation Association, 2009.

[54] X. Fu, T. Li, and J. A. Fortes. Architecting reliable multi-core network-on-chip for small scale processing technology. In *2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*, pages 111–120. IEEE, 2010.

[55] Z. Ghaderi, A. Alqahtani, and N. Bagherzadeh. Online monitoring and adaptive routing for aging mitigation in nocs. In *2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 67–72. IEEE, 2017.

[56] Z. Ghaderi and E. Bozorgzadeh. Aging-aware high-level physical planning for reconfigurable systems. In *Design Automation Conference (ASP-DAC), 2016 21st Asia and South Pacific*, pages 631–636. IEEE, 2016.

[57] Z. Ghaderi, M. Ebrahimi, Z. Navabi, E. Bozorgzadeh, and N. Bagherzadeh. Sensible: A highly scalable sensor design for path-based age monitoring in fpgas. *IEEE Transactions on Computers*, 66(5):919–926, 2017.

[58] Z. Ghaderi, M. Ebrahimi, Z. Navabi, E. Bozorgzadeh, and N. Bagherzadeh. Sensible: A highly scalable sensor design for path-based age monitoring in fpgas. *IEEE Transactions on Computers*, 66(5):919–926, 2017.

[59] C. J. Glass and L. M. Ni. Adaptive routing in mesh-connected networks. In *Distributed Computing Systems, 1992., Proceedings of the 12th International Conference on*, pages 12–19. IEEE, 1992.

[60] C. J. Glass and L. M. Ni. The turn model for adaptive routing. *ACM SIGARCH Computer Architecture News*, 20(2):278–287, 1992.

[61] C. J. Glass and L. M. Ni. The turn model for adaptive routing. *ACM SIGARCH Computer Architecture News*, 20(2):278–287, 1992.

[62] B. Goplen and S. Sapatnekar. Thermal via placement in 3d ics. In *Proceedings of the 2005 international symposium on Physical design*, pages 167–174. ACM, 2005.

[63] B. Goplen and S. S. Sapatnekar. Placement of thermal vias in 3-d ics using various thermal objectives. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(4):692–709, 2006.

[64] P. Gratz, B. Grot, and S. W. Keckler. Regional congestion awareness for load balance in networks-on-chip. In *High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on*, pages 203–214. IEEE, 2008.

[65] D. Hasselman and L. F. Johnson. Effective thermal conductivity of composites with interfacial thermal barrier resistance. *Journal of Composite Materials*, 21(6):508–515, 1987.

[66] R. Ho, K. W. Mai, and M. A. Horowitz. The future of wires. *Proceedings of the IEEE*, 89(4):490–504, 2001.

[67] P.-Y. Hsu, H.-T. Chen, and T. Hwang. Stacking signal tsv for thermal dissipation in global routing for 3-d ic. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(7):1031–1042, 2014.

[68] W.-H. Hu, S. E. Lee, and N. Bagherzadeh. Dmesh: a diagonally-linked mesh network-on-chip architecture. *Network on Chip Architectures*, page 14, 2008.

[69] L. Huang, F. Yuan, and Q. Xu. On task allocation and scheduling for lifetime extension of platform-based mpsoc designs. *IEEE Transactions on Parallel and Distributed Systems*, 22(12):2088–2099, 2011.

[70] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan. Hotspot: A compact thermal modeling methodology for early-stage vlsi design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 14(5):501–513, 2006.

[71] V. Huard, M. Denais, and C. Parthasarathy. Nbti degradation: From physical mechanisms to modelling. *Microelectronics Reliability*, 46(1):1–23, 2006.

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

[73] T. Jiang, J. Im, R. Huang, and P. S. Ho. Through-silicon via stress characteristics and reliability impact on 3d integrated circuits. *MRS Bulletin*, 40(3):248–256, 2015.

[74] S. G. Kandlikar. Review and projections of integrated cooling systems for three-dimensional integrated circuits. *Journal of Electronic Packaging*, 136(2):024001, 2014.

[75] J. Keane, X. Wang, D. Persaud, and C. H. Kim. An all-in-one silicon odometer for separately monitoring hci, bti, and tddb. *IEEE Journal of Solid-State Circuits*, 45(4):817–829, 2010.

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

[77] D. H. Kim, K. Athikulwongse, and S. K. Lim. Study of through-silicon-via impact on the 3-d stacked ic layout. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 21(5):862–874, 2013.

[78] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, V. Narayanan, M. S. Yousif, and C. R. Das. A novel dimensionally-decomposed router for on-chip communication in 3d architectures. In *ACM SIGARCH Computer Architecture News*, volume 35, pages 138–149. ACM, 2007.

[79] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, V. Narayanan, M. S. Yousif, and C. R. Das. A novel dimensionally-decomposed router for on-chip communication in 3d architectures. In *ACM SIGARCH Computer Architecture News*, volume 35, pages 138–149. ACM, 2007.

[80] J. Kim, C. Nicopoulos, D. Park, V. Narayanan, M. S. Yousif, and C. R. Das. A gracefully degrading and energy-efficient modular router architecture for on-chip networks. *ACM SIGARCH Computer Architecture News*, 34(2):4–15, 2006.

[81] J. U. Knickerbocker, P. Andry, B. Dang, R. Horton, C. S. Patel, R. Polastre, K. Sakuma, E. Sprogis, C. Tsang, B. Webb, et al. 3d silicon integration. In *Electronic Components and Technology Conference, 2008. ECTC 2008. 58th*, pages 538–543. IEEE, 2008.

[82] C.-T. Ko and K.-N. Chen. Wafer-level bonding/stacking technology for 3d integration. *Microelectronics reliability*, 50(4):481–488, 2010.

[83] A. Kohler, G. Schley, and M. Radetzki. Fault tolerant network on chip switching with graceful performance degradation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(6):883–896, 2010.

[84] R. Kumar and G. Hinton. A family of 45nm ia processors. In *Solid-State Circuits Conference-Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pages 58–59. IEEE, 2009.

[85] J. H. Lau. Evolution, challenge, and outlook of tsv, 3d ic integration and 3d silicon integration. In *Advanced Packaging Materials (APM), 2011 International Symposium on*, pages 462–488. IEEE, 2011.

[86] J. H. Lau and T. G. Yue. Thermal management of 3d ic integration with tsv (through silicon via). In *Electronic Components and Technology Conference, 2009. ECTC 2009. 59th*, pages 635–640. IEEE, 2009.

[87] C.-R. Li, W.-K. Mak, and T.-C. Wang. Fast fixed-outline 3-d ic floorplanning with tsv co-placement. *IEEE transactions on very large scale integration (VLSI) systems*, 21(3):523–532, 2013.

[88] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi. Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 469–480. ACM, 2009.

[89] Z. Li, X. Hong, Q. Zhou, S. Zeng, J. Bian, H. Yang, V. Pitchumani, and C.-K. Cheng. Integrating dynamic thermal via planning with 3d floorplanning algorithm. In *Proceedings of the 2006 international symposium on Physical design*, pages 178–185. ACM, 2006.

[90] Z. Liu, S. Swarup, S. X.-D. Tan, H.-B. Chen, and H. Wang. Compact lateral thermal resistance model of tsvs for fast finite-difference based thermal analysis of 3-d stacked ics. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(10):1490–1502, 2014.

[91] G. H. Loh. 3d-stacked memory architectures for multi-core processors. In *ACM SIGARCH computer architecture news*, volume 36, pages 453–464. IEEE Computer Society, 2008.

[92] D. Lorenz, G. Georgakos, and U. Schlichtmann. Aging analysis of circuit timing considering nbti and hci. In *On-Line Testing Symposium, 2009. IOLTS 2009. 15th IEEE International*, pages 3–8. IEEE, 2009.

[93] N. Madan and R. Balasubramonian. Leveraging 3d technology for improved reliability. In *Microarchitecture, 2007. MICRO 2007. 40th Annual IEEE/ACM International Symposium on*, pages 223–235. IEEE, 2007.

[94] A. Mejia, M. Palesi, J. Flich, S. Kumar, P. López, R. Holsmark, and J. Duato. Region-based routing: a mechanism to support efficient routing algorithms in nocs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 17(3):356–369, 2009.

[95] J. Meng, K. Kawakami, and A. K. Coskun. Optimizing energy efficiency of 3-d multicore systems with stacked dram under power and thermal constraints. In *Proceedings of the 49th Annual Design Automation Conference*, pages 648–655. ACM, 2012.

[96] P. Mercati, F. Paterna, A. Bartolini, L. Benini, and T. Rosing. Warm: Workload-aware reliability management in linux/android. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2016.

[97] E. Mintarno, J. Skaf, R. Zheng, J. Velamala, Y. Cao, S. Boyd, R. W. Dutton, and S. Mitra. Optimized self-tuning for circuit aging. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 586–591. European Design and Automation Association, 2010.

[98] T. R. Mück, Z. Ghaderi, N. D. Dutt, and E. Bozorgzadeh. Exploiting heterogeneity for aging-aware load balancing in mobile platforms. *IEEE Transactions on Multi-Scale Computing Systems*, 3(1):25–35, 2017.

[99] T. R. Mück, Z. Ghaderi, N. D. Dutt, and E. Bozorgzadeh. Exploiting heterogeneity for aging-aware load balancing in mobile platforms. *IEEE Transactions on Multi-Scale Computing Systems*, 3(1):25–35, 2017.

[100] C.-W. Nan, R. Birringer, D. R. Clarke, and H. Gleiter. Effective thermal conductivity of particulate composites with interfacial thermal resistance. *Journal of Applied Physics*, 81(10):6692–6699, 1997.

[101] L. M. Ni and P. K. McKinley. A survey of wormhole routing techniques in direct networks. *Computer*, 26(2):62–76, 1993.

[102] S. Ogawa and N. Shiono. Generalized diffusion-reaction model for the low-field charge-buildup instability at the si-sio 2 interface. *Physical Review B*, 51(7):4218, 1995.

[103] M. Palesi, R. Holsmark, S. Kumar, and V. Catania. Application specific routing algorithms for networks on chip. *IEEE Transactions on Parallel and Distributed Systems*, 20(3):316–330, 2009.

[104] A. Patel, F. Afram, S. Chen, and K. Ghose. Marss: a full system simulator for multicore x86 cpus. In *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, pages 1050–1055. IEEE, 2011.

[105] F. Paterna, A. Acquaviva, and L. Benini. Aging-aware energy-efficient workload allocation for mobile multimedia platforms. *IEEE Transactions on Parallel and Distributed Systems*, 24(8):1489–1499, 2013.

[106] M. Pathak and S. K. Lim. Performance and thermal-aware steiner routing for 3-d stacked ics. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 28(9):1373–1386, 2009.

[107] H. N. Phan and D. Agonafer. Experimental analysis model of an active cooling method for 3d-ics utilizing multidimensional configured thermoelectric coolers. *Journal of Electronic Packaging*, 132(2):024501, 2010.

[108] K. Pietrak and T. S. Wiśniewski. A review of models for effective thermal conductivity of composite materials. *Journal of Power Technologies*, 95(1):14–24, 2014.

[109] K. Puttaswamy and G. H. Loh. Thermal analysis of a 3d die-stacked high-performance microprocessor. In *Proceedings of the 16th ACM Great Lakes symposium on VLSI*, pages 19–24. ACM, 2006.

[110] Y. Qian, Z. Lu, and W. Dou. From 2d to 3d nocs: a case study on worst-case communication performance. In *Computer-Aided Design-Digest of Technical Papers, 2009. ICCAD 2009. IEEE/ACM International Conference on*, pages 555–562. IEEE, 2009.

[111] P. Ren, M. Lis, M. H. Cho, K. S. Shim, C. W. Fletcher, O. Khan, N. Zheng, and S. Devadas. Hornet: A cycle-level multicore simulator. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 31(6):890–903, 2012.

[112] Z. Ren and J. Lee. Thermal conductivity anisotropy in holey silicon nanostructures and its impact on thermoelectric cooling. *Nanotechnology*, 29(4):045404, 2017.

[113] Z. Ren, Z. Yu, J. C. Kim, and J. Lee. Tsv-integrated thermoelectric cooling by holey silicon for hot spot thermal management. *Nanotechnology*, 30(3):035201, 2018.

[114] S. Rodrigo, S. Medardoni, J. Flich, D. Bertozzi, and J. Duato. Efficient implementation of distributed routing algorithms for nocs. *IET Computers & Digital Techniques*, 3(5):460–475, 2009.

[115] R. Salamat, M. Khayambashi, M. Ebrahimi, and N. Bagherzadeh. A resilient routing algorithm with formal reliability analysis for partially connected 3d-nocs. *IEEE Transactions on Computers*, 65(11):3265–3279, 2016.

[116] D. Sanchez and C. Kozyrakis. Zsim: Fast and accurate microarchitectural simulation of thousand-core systems. In *ACM SIGARCH Computer architecture news*, volume 41, pages 475–486. ACM, 2013.

[117] T. Schonwald, J. Zimmermann, O. Bringmann, and W. Rosenstiel. Fully adaptive fault-tolerant routing algorithm for network-on-chip architectures. In *Digital System Design Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference on*, pages 527–534. IEEE, 2007.

[118] D. Sengupta and S. Sapatnekar. Estimating circuit aging due to bti and hci using ring-oscillator-based sensors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017.

[119] Y. Shang, C. Zhang, H. Yu, C. S. Tan, X. Zhao, and S. K. Lim. Thermal-reliable 3d clock-tree synthesis considering nonlinear electrical-thermal-coupled tsv model. In *2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 693–698. IEEE, 2013.

[120] B. Shi, A. Srivastava, and A. Bar-Cohen. Hybrid 3d-ic cooling system using microfluidic cooling and thermal tsvs. In *2012 IEEE Computer Society Annual Symposium on VLSI*, pages 33–38. IEEE, 2012.

[121] K. S. Shim, M. H. Cho, M. Kinsy, T. Wen, M. Lis, G. E. Suh, and S. Devadas. Static virtual channel allocation in oblivious routing. In *Proceedings of the 2009 3rd ACM/IEEE International Symposium on Networks-on-Chip*, pages 38–43. IEEE Computer Society, 2009.

[122] A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunschwiler, and D. Atienza. 3d-ice: Fast compact transient thermal modeling for 3d ics with inter-tier liquid cooling. In *2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 463–470. IEEE, 2010.

[123] E. E. Swartzlander. Parallel counters. *IEEE Transactions on computers*, 100(11):1021–1024, 1973.

[124] E. Takeda and N. Suzuki. An empirical model for device degradation due to hot-carrier injection. *IEEE electron device letters*, 4(4):111–113, 1983.

[125] M. Tang, X. Lin, and M. Palesi. An offline method for designing adaptive routing based on pressure model. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(2):307–320, 2015.

[126] A. Tiwari and J. Torrellas. Facelift: Hiding and slowing down aging in multicores. In *Proceedings of the 41st annual IEEE/ACM International Symposium on Microarchitecture*, pages 129–140. IEEE Computer Society, 2008.

[127] V. M. van Santen, H. Amrouch, J. Martin-Martinez, M. Nafria, and J. Henkel. Designing guardbands for instantaneous aging effects. In *Design Automation Conference (DAC), 2016 53nd ACM/EDAC/IEEE*, pages 1–6. IEEE, 2016.

[128] V. Venkatadri, B. Sammakia, K. Srihari, and D. Santos. A review of recent advances in thermal management in three dimensional chip stacks in electronic systems. *Journal of Electronic packaging*, 133(4):041011, 2011.

[129] M. M. Waldrop. The chips are down for moores law. *Nature News*, 530(7589):144, 2016.

[130] J. Wang, J. Beu, R. Bheda, T. Conte, Z. Dong, C. Kersey, M. Rasquinha, G. Riley, W. Song, H. Xiao, et al. Manifold: A parallel simulation framework for multicore systems. In *Performance Analysis of Systems and Software (ISPASS), 2014 IEEE International Symposium on*, pages 106–115. IEEE, 2014.

[131] L. Wang, X. Wang, and T. Mak. Dynamic programming-based lifetime aware adaptive routing algorithm for network-on-chip. In *Very Large Scale Integration (VLSI-SoC), 2014 22nd International Conference on*, pages 1–6. IEEE, 2014.

[132] W. Wang, S. Yang, S. Bhardwaj, S. Vrudhula, F. Liu, and Y. Cao. The impact of nbti effect on combinational circuit: modeling, simulation, and analysis. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 18(2):173–183, 2010.

[133] E. Wong and S. K. Lim. 3d floorplanning with thermal vias. In *Proceedings of the Design Automation & Test in Europe Conference*, volume 1, pages 1–6. IEEE, 2006.

[134] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta. The splash-2 programs: Characterization and methodological considerations. In *ACM SIGARCH computer architecture news*, volume 23, pages 24–36. ACM, 1995.

[135] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta. The splash-2 programs: Characterization and methodological considerations http://www-flash.standford.edu/apps/splash. *ACM SIGARCH computer architecture news*, 23(2):24–36, 1995.

[136] 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, 2016.

[137] A. Zeng, J. Lu, K. Rose, and R. J. Gutmann. First-order performance prediction of cache memory with wafer-level 3d integration. *IEEE Design & Test of Computers*, 22(6):548–555, 2005.

[138] R. Zhang, K. Roy, C.-K. Koh, and D. B. Janes. Power trends and performance characterization of 3-dimensional integration. In *ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No. 01CH37196)*, volume 4, pages 414–417. IEEE, 2001.

[139] R. Zhang, M. R. Stan, and K. Skadron. Hotspot 6.0: Validation, acceleration and extension. *University of Virginia, Tech. Rep*, 2015.

[140] Y. Zhang and M. Bakir. Independent interlayer microfluidic cooling for heterogeneous 3d ic applications. *Electronics Letters*, 49(6):404–406, 2013.

[141] D. Zhao and G. Tan. A review of thermoelectric cooling: materials, modeling and applications. *Applied Thermal Engineering*, 66(1-2):15–24, 2014.

[142] Y. Zhao, C. Hao, and T. Yoshimura. Tsv assignment of thermal and wirelength optimization for 3d-ic routing. In *2018 28th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pages 155–162. IEEE, 2018.

[143] C. Zhu, Z. Gu, L. Shang, R. P. Dick, and R. Joseph. Three-dimensional chip-multiprocessor run-time thermal management. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(8):1479–1492, 2008.