PHYSICAL PARAMETER-AWARE NETWORKS-ON-CHIP DESIGN

Nizar Saadi Dahir

A Thesis Submitted for the Degree of
Doctor of Philosophy at Newcastle University

School of Electrical and Electronic Engineering
Faculty of Science, Agriculture and Engineering
February 2015

DECLARATION

I hereby declare that this thesis is my own work and effort and that it has not been submitted anywhere for any award. Where other sources of information have been used, they have been acknowledged.

Newcastle upon Tyne February 2015

NIZAR DAHIR

CERTIFICATE OF APPROVAL

I confirm that, to the best of my knowledge, this thesis is from the student's own work and effort, and all other sources of information used have been acknowledged. This thesis has been submitted with my approval.

ALEX YAKOVLEV

To my beloved wife, Noor, and our sweethearts, Warqaa and Zaid.
— Nizar

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my supervisors Prof. Alex Yakovlev, Dr Terrence Mak and Dr Fei Xia for their support, inspiring comments and guidance through my studies.

I am also grateful to the members of the Microelectronics System Design group at Newcastle University for their assistance and guidance in my studies. I appreciate the help of my friends and colleagues, Ra'ed Aldujaily, Ghaith Tarawneh, and Petros Missailidis for inspiring and wonderful discussions, productive criticism and suggestions, and for being very nice friends and supporters.

I am also grateful to Dr Maurizio Palesi for his advice and suggestions in modifying the NoC simulator and power model.

I would like to offer my special regards to all the staff of the school of Electrical and Electronic Engineering.

My father is also owed many thanks for his continuous support and encouragement and backup from abroad, which has helped me to overcome many difficulties and has given me strong motivation throughout my studies.

ABSTRACT

Networks-on-Chip (NoCs) have been proposed as a scalable, reliable and power-efficient communication fabric for chip multiprocessors (CMPs) and multiprocessor systems-on-chip (MPSoCs). NoCs determine both the performance and the reliability of such systems, with a significant power demand that is expected to increase due to developments in both technology and architecture. In terms of architecture, an important trend in many-core systems architecture is to increase the number of cores on a chip while reducing their individual complexity. This trend increases communication power relative to computation power. Moreover, technology-wise, power-hungry wires are dominating logic as power consumers as technology scales down. For these reasons, the design of future very large scale integration (VLSI) systems is moving from being computation-centric to communication-centric.

On the other hand, chip's physical parameters integrity, especially power and thermal integrity, is crucial for reliable VLSI systems. However, guaranteeing this integrity is becoming increasingly difficult with the higher scale of integration due to increased power density and operating frequencies that result in continuously increasing temperature and voltage drops in the chip. This is a challenge that may prevent further shrinking of devices. Thus, tackling the challenge of power and thermal integrity of future many-core systems at only one level of abstraction, the chip and package design for example, is no longer sufficient to ensure the integrity of physical parameters. New design-time and run-time strategies may need to work together at different levels of abstraction, such as package, application, network, to provide the required physical parameter integrity for these large systems. This necessitates strategies that work at the level of the on-chip network with its rising power budget.

This thesis proposes models, techniques and architectures to improve power and thermal integrity of Network-on-Chip (NoC)-based many-core systems. The thesis is composed of two major parts: i) minimization and modelling of power supply variations to improve power integrity; and ii) dynamic thermal adaptation to improve thermal integrity. This thesis makes four major contributions. The first is a computational model of on-chip power supply variations in NoCs. The proposed model embeds a power delivery model, an NoC activity simulator and a power model. The model is verified with SPICE simulation and employed to analyse power supply variations in synthetic and real NoC workloads. Novel observations regarding power supply noise correlation with different traffic patterns and routing algorithms are found. The second is a new application mapping strategy aiming

to minimize power supply noise in NoCs. This is achieved by defining a new metric, switching activity density, and employing a force-based objective function that results in minimizing switching density. Significant reductions in power supply noise (PSN) are achieved with a low energy penalty. This reduction in PSN also results in a better link timing accuracy. The third contribution is a new dynamic thermal-adaptive routing strategy to effectively diffuse heat from the NoC-based three-dimensional (3D) CMPs, using a dynamic programming (DP)-based distributed control architecture. Moreover, a new approach for efficient extension of two-dimensional (2D) partially-adaptive routing algorithms to 3D is presented. This approach improves three-dimensional network-on-chip (3D NoC) routing adaptivity while ensuring deadlock-freeness. Finally, the proposed thermal-adaptive routing is implemented in field-programmable gate array (FPGA), and implementation challenges, for both thermal sensing and the dynamic control architecture are addressed. The proposed routing implementation is evaluated in terms of both functionality and performance.

The methodologies and architectures proposed in this thesis open a new direction for improving the power and thermal integrity of future NoC-based 2D and 3D many-core architectures.

PUBLICATIONS

Journal publications:

1. **N. Dahir**, T. Mak, F. Xia, and A. Yakovlev, *Modelling and Tools for Power Supply Variations Analysis in Networks-on-Chip*, IEEE Transactions on Computers (TC), vol. 99, PrePrints, pp. 1-14, 20 Nov. 2012..

2. **N. Dahir**, R. Al-Dujaily, T. Mak, and A. Yakovlev, *Thermal Optimization in Network-on-Chip Based 3D Chip Multiprocessors Using Dynamic Programming Networks*, ACM Transactions on Embedded Computing Systems (TECS), (To appear), pp. 1-25, 2013.

3. **N. Dahir**, T. Mak, R. Al-Dujaily, and A. Yakovlev, *Highly Adaptive and Deadlock-Free Routing for Three-Dimensional Networks-on-Chip*, Institution of Engineering and Technology (IET), Computers & Digital Techniques (CDT), 7:255-263, Nov. 2013.

4. **N. Dahir**, T. Mak, F. Xia, and A. Yakovlev, *Power Supply Noise Minimization by Activity-Aware Mapping in Networks-on-Chip*, (revision submitted), IEEE Transactions on Parallel and Distributed Systems (TPDS), pp. 1-14, 2013.

5. R. Al-Dujaily, **N. Dahir**, T. Mak, F. Xia, and A. Yakovlev. *Dynamic Programming-based Runtime Thermal Management (DPRTM): An On-line Thermal Control Strategy for 3D-NoC Systems*, ACM Transactions on Design Automation of Electronic Systems (TODAES), vol. 19, issue 1, article 2, pp. 1-28, December 2013.

Conference publications:

1. **N. Dahir**, T. Mak, and A. Yakovlev, *Communication Centric On-Chip Power Grid Models for Networks-on-Chip*, In VLSI and System-on-Chip (VLSI-SoC), 2011 IEEE/IFIP 19th International Conference on, pp. 180-183, 2011.

2. **N. Dahir**, T. Mak, F. Xia, and A. Yakovlev, *Minimizing Power Supply Noise Through Harmonic Mappings in Networks-on-Chip*, In Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis, CODES+ISSS'12, pp. 113-122, New York, NY, USA, 2012.

3. **N. Dahir**, G. Tarawneh, T. Mak, R. Al-Dujaily and A. Yakovlev, *Design and Implementation of Dynamic Thermally-Adaptive Routing Strategy for Networks-on-Chip*, In Parallel, Distributed and Network-Based Processing (PDP), 2014 22nd Euromicro International Conference on, Turin-Italy, 12-14 Feb. 2014.

Workshop and forum publications:

1. **N. Dahir**, R. Al-Dujaily, A. Yakovlev, P. Missailidis, and T. Mak. *Deadlock-free and plane-balanced adaptive routing for 3D networks-on-chip*, In Proceedings of the Fifth International Workshop on Network on Chip Architectures, NoCArc'12, pages 31-36, New York, NY, USA, 2012. ACM.

2. **N. Dahir**, T. Mak and A. Yakovlev, *Adaptive Run-Time Thermal Balancing in 3D Network-on-Chip Systems Using Dynamic Programming Networks*, Proceedings of UK Electronics Forum (UKEF'12), Newcastle, 30-31th Aug. 2012, pp. 1 - 6.

3. **N. Dahir**, T. Mak and A. Yakovlev, *Networks-On-Chip Workload Impact on Power Delivery Grid*, Proceedings of UK Electronics Forum (UKEF'11), Manchester, 4-5th July, 2011, pp. 6 - 12.

I also contributed in the following works:

1. A. Karkar, **N. Dahir**, K. Tong, R. Al-Dujaily, A. Yakovlev, and T. Mak, *Hybrid Wire-Surface Wave Architecture for One-to-Many Communication in Network-on-Chip*, In Proceedings of the conference on Design, automation and test in Europe, DATE'14, Dresden-Germany, 24-28 Mar. 2014.

2. H. Alrudainy, **N. Dahir**, A. Mokhov, A. Yakovlev, *A Scalable Physical Model for Nano-Electro-Mechanical Relays*, Design automation conference, DAC'14, San Francisco, CA, 1-5 June 2014

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACRONYMS

2D two-dimensional

3D NoCs three-dimensional networks-on-chip

3D NoC three-dimensional network-on-chip

3D three-dimensional

3D-ICs three-dimensional integrated circuits

ASIC application-specific integrated circuit

BE best effort

BER bit error rate

CMP chip-multiprocessor

CMPs chip multiprocessors

DOR dimension-ordered routing

DP dynamic programming

DPN dynamic programming network

DSP digital signal processing

DVFS dynamic voltage and frequency scaling

FIFO First-In-First-Out

FPGA field-programmable gate array

GALS globally asynchronous locally synchronous

GT guaranteed service

ILP instruction-level parallelism

IP intellectual property

IR resistive

MIQP mixed integer quadratic programming

MPSoC multiprocessor system-on-chip

MPSoCs multiprocessor systems-on-chip

MTTF mean-time-to-failure

NI network interface

NIs network interfaces

NoC Network-on-Chip

NoCs Networks-on-Chip

NoP Neighbours-on-Path

OCP Open Core Protocol

OCP-IP open core protocol-international partnership

PE processing element

PEs processing elements

PSN power supply noise

QoS quality of service

RF radio frequency

RO ring oscillator

RTM run-time thermal management

SA simulated annealing

SoC system-on-chip

SoCs systems-on-chip

STA statistical timing analysis

TSVs through-silicon vias

VCs virtual channels

VLSI very large scale integration

MOR model order reduction

PDN power delivery network

PIR packet injection rate

FIT failures-in-time

Part I

Thesis Chapters

1 INTRODUCTION

1.1 MOTIVATION

Continuing technology scaling is enabling the integration of billions of gates in a chip, and Moore's law is expected to hold for the next fifteen years. This rapidly increasing integration density allows hundreds to thousands of intellectual property (IP) cores to be placed in one chip [103]. Placing that many IP cores in one chip comes with many challenges. The major challenge is to provide an efficient and reliable communication fabric among these cores. This communication fabric must provide scalable, reliable and power-efficient on-chip communication. To cope with the communication requirements of these many-core architectures, Network-on-Chip (NoC) is proposed as a modular packet-switched communication paradigm for many-core very large scale integration (VLSI) architectures.

In system-on-chip (SoC), multiprocessor system-on-chip (MPSoC) and chip-multiprocessor (CMP), Networks-on-Chip (NoCs) can tackle many limitations associated with traditional bus-based on-chip interconnections [23]. Moreover, NoCs can provide IP re-usability and the standardization of communication interfaces, such as in the open core protocol-international partnership (OCP-IP) protocol. These characteristics are crucial in delivering industry-standard flexibility and to facilitate the plug and play IP integration which reduces design effort and the time-to-market for future VLSI systems. Notable examples of architectures that have adopted NoCs are Intel's 80-core TeraFLOPS [193], Tilera's TILE64 [183], and MIT's RAW chip [181].

The power budget of NoCs can represent a significant portion of overall chip power and this portion is expected to increase in the future [147]. There are two main reasons for this; technology scaling and the trends in many-core architecture. In terms of technology scaling, there is a significant difference in scaling between metal and silicon. Smaller feature size would cause interconnects to dominate logic in power consumption [104]. This is mainly because shrinking technology node size results in lower power consumption and delay for logic gates whereas interconnects become relatively slower and more power hungry. As a result, interconnect power would take up the majority of total chip power in future technology nodes. Moreover, the majority of interconnects power is dissipated by global interconnects [147, 127], which are part of NoCs.

In terms of architecture, the International Technology Roadmap for Semiconductors (ITRS) predicts that the number of cores that can be

placed in a chip will exponentially increase with technology scaling [104], as shown in Fig. 1.1. Moreover, a trend in many-core system microarchitectures favours the integration of many (hundreds or thousands) simple cores over few complex cores [33]. This results in higher performance and provides finer control on these cores with dynamic voltage and frequency scaling (DVFS). However, this will result in lower core complexity and higher core number, which increases communication power consumption relative to computation power. On one hand, this is due to the simplicity of these cores, while on the other hand, a higher number of connected cores increases network communication activity. This is exacerbated by the fact that power management in the network is not preferred since such techniques can lead to a wake up latency which can severely affect performance [33].

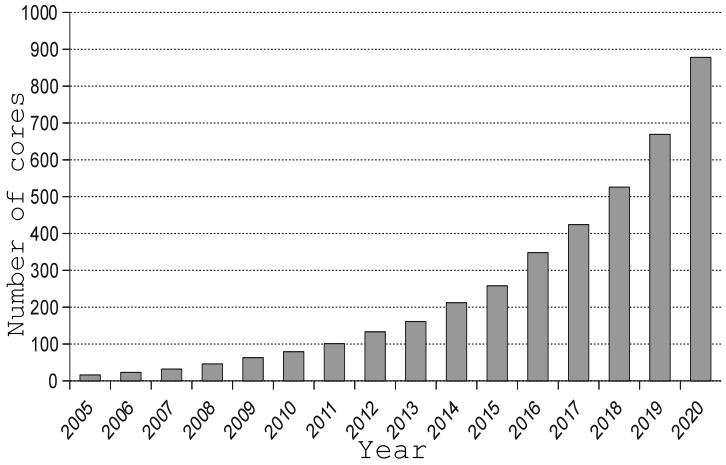


Figure 1.1: Expected number of processing elements in a SoC [103].

As a result of the aforementioned factors, many-core VLSI design is moving towards being communication-centric and NoC power dissipation is becoming increasingly significant. Thus, NoCs are responsible not only for higher energy consumption, but also for the unwanted effects associated with high energy dissipation and power density, such as power supply variations, or power supply noise (PSN), and thermal noise. As a result, the modelling, characterization and management of the workload of NoCs taking into account their PSN and thermal impacts in modern many-core systems is becoming increasingly indispensable. This is particularly important for applications with high communication demand, where the network is shown to surpass the cores in varying these physical parameters [170].

High power consumption density is a source of both increased PSN and high temperature in VLSI systems. Moreover, PSN and high temperature are exacerbated by technology scaling due to increased device and switching activity densities [35, 154, 17, 39]. PSN and temperature can severely degrade the performance and reliability of VLSI systems. However, the most prominent impact of PSN is on the delay

of both logic and interconnects [22, 43, 163, 116]. High temperatures can also increase leakage and delay [154, 96]. However, a major impact of high temperature in VLSI systems is to reduce lifetime reliability due to reduced mean-time-to-failure (MTTF) [35, 175]. High temperature is a major challenge for the emerging three-dimensional (3D) system integration or three-dimensional integrated circuits (3D-ICs) in particular.

In 3D system integration, multiple die (or wafer) layers are stacked vertically and connected using through-silicon vias (TSVs). This technology is promising and can increase integration capacity and result in less global interconnects length and delay. In particular, with many-core system integration and NoCs, 3D integration enables a massive integration of cores in a chip and 3D NoCs are ideal for connecting these cores [204, 167]. 3D NoCs can provide high data bandwidth due to their shorter interconnects and lower average hop counts [204, 167]. However, 3D integration comes with many challenges, of which thermal challenge is the most alerting due to the high power density and longer major heat diffusion paths [126, 38, 86].

A considerable amount of research has been conducted in the field of the modelling and mitigation of both PSN and thermal noise for CMP and MPSoC systems [75, 65, 126, 209, 17, 163, 185]. However, most studies assume applications with independent tasks and the impact of the rapidly increasing NoC communication workload is often ignored. Moreover, the impact of PSN and thermal noise on NoC performance and reliability is rarely studied.

This thesis focuses on the impact of NoC workload on PSN and temperature in multi/many core VLSI systems. In contrast to processors, this workload would have an interesting correlation with the temporal and spatial distributions of traffic load. NoC workload can be determined in early design stages, once the characteristics of the application are known. Moreover, NoC workload is spread across most of the chip area. Thus, by controlling this workload, a considerable overall mitigation of PSN and temperature noise can be achieved.

1.2 THESIS CONTRIBUTIONS

The major contributions of this thesis can be summarized as follows:

- Developing a tool for computing the on-chip power supply noise caused by NoC traffic patterns. This tool integrates models for the power grid, NoC traffic simulation, and router microarchitectural power. Moreover, rigorous verification of the models is conducted using SPICE simulations. The developed models have been employed to analyse the power supply noise in NoCs. Novel correlations are found between PSN and different traffic patterns and routing algorithms [54, 51].

- Studying the impact of the resulting PSN on NoC reliability. A statistical timing analysis (STA) of link timing is conducted in the presence of power supply noise. From this analysis high level fault metrics are evaluated such as the probability of timing errors and bit error rate (BER) for both real world and synthetic communication scenarios. Correlations between these metrics and NoC traffic patterns, routing algorithms and application task mapping are then analysed and discussed [54, 53]

- A new concept for optimizing PSN in NoCs application mapping is introduced. The proposed mapping considers the impact of communication workload on the power delivery network in multi-core NoC-based systems using the new metric of *activity density* and the analysis of the impact of its spatial patterns on power supply integrity. Relationships between the spatial distribution patterns of core activity and PSN are studied and discussed [53].

- *Tile repulsive force* is proposed as an objective of mapping strategy which, in contrast to other NoC mapping strategies, results in spreading high activity tiles across the chip in order to minimize activity density. This achieves significant reductions in PSN with low energy penalties compared to energy mapping. Moreover, statistical timing analysis of the resulting systems shows a considerable reduction in BER. This is achieved in the new mapping strategy due to the reduced PSN, and the lower frequency of timing violations, which leads to better timing accuracy [53].

- Proposing a new run-time thermal-adaptive routing strategy which can effectively diffuse heat from NoC-based 3D chip multiprocessors (CMPs). This strategy uses a distributed dynamic programming-based control architecture called the dynamic programming network (DPN). Moreover, the DPN is improved such that computation and propagation of the cost is in compliance with deadlock-free routing algorithms. The proposed routing is evaluated through experimental studies and comparisons with state-of-the-art NoC run-time thermal management (RTM) schemes using various synthetic and real traffic scenarios. Temperature, reliability, energy and performance results are compared and discussed. Moreover, the hardware implementation of the proposed method is discussed in detail and area and power overheads are evaluated. The proposed technique is shown to outperform existing RTM approaches in terms of thermal regulation, chip reliability and performance [55, 14].

- A new approach for extending two-dimensional (2D) partially-adaptive routing algorithms to 3D is introduced. This improves the adaptive 3D NoC routing algorithms in order to achieve higher

path diversity and more balanced adaptiveness. As a result, better DPN performance is gained. This is achieved by applying different turn prohibition rules for different layers, resulting in different restrictions on traffic flow for different layers and a more balanced degree of adaptiveness [55, 56, 52].

- The DPN-based RTM for NoCs is implemented in field-programmable gate array (FPGA). Direct on-chip temperature readings from distributed thermal sensors are used, and a low-cost implementation of DPNis achieved. For thermal sensing, ring oscillator (RO) is used and the challenges associated with sensor accuracy and precision, including V_{DD} drop isolation and compensation for intra-chip process variations, are addressed. In terms of functionality, the proposed design is shown to be highly flexible in maneuvering packets away from hot regions. This results in reductions in maximum chip temperature up to 16% and chip's thermal gradient is reduced by up to 51% compared to performance-driven routing. Moreover, the proposed scheme results in significantly slower chip heating, reflected in higher performance of up to 100% when the chip works under a thermal limit [57].

1.3 THESIS LAYOUT

The thesis is organised into seven chapters. The major contributions of the thesis are described in two major parts. The modelling and optimization of PSN in NoCs is covered in Chapters 3 and 4, while Chapters 5 and 6 cover thermal optimization in NoCs. The chapters are summarized as follows:

Chapter 1 "Introduction". Introduces the motivations, contributions, assumptions and layout of the thesis.

Chapter 2 "Background and Literature Review". Describes the background of relevant theory and applications concerning on-chip interconnection networks. Also, a review of the methodologies for NoC research and industry designs is presented.

Chapter 3 "Power Supply Variations Modelling in Networks-on-Chip". Explains a modelling tool of power supply variations (noise) in NoCs. This tool integrates a fast power grid model, an NoC traffic simulator, an on-chip link model and a router energy model. The chapter also demonstrates the use of the proposed model in analysing the impact of PSN on the reliability of NoC interconnects. This is conducted through a STA of NoC interconnects in the presence of power supply variations and the evaluation of BER.

Chapter 4 "Power Supply Noise Minimization Mapping". A new mapping strategy is proposed in this chapter. This mapping aims for activity balancing in the chip, which is achieved by employing a force-based metric optimization. Metrics for regional activity density

are defined and their impacts on PSN are analysed. Evaluation results of the proposed strategy, with many real-application benchmarks, are presented and discussed.

Chapter 5 "Dynamic Thermal Optimization in 3D NoCs". An adaptive run-time routing strategy is introduced. This strategy effectively optimizes heat distribution in 3D NoC-based CMPs. This is achieved by employing DPN to select and optimize the direction of data maneuver in the NoC with thermal-awareness. Moreover, a technique for improving routing algorithms path diversity and degree of adaptiveness is presented and evaluated. The proposed routing scheme is compared with recent NoCs thermal optimization techniques in terms of thermal reduction, reliability improvement, and throughput performance overhead.

Chapter 6 "FPGA Implementation of Thermal-Adaptive Routing in NoCs". In this chapter, FPGA implementation of the proposed thermal-adaptive routing in NoCs is described. DPN is used to implement the adaptive routing control logic and ROs are used for temperature sensing implementation. Challenges associated with DPN and sensor implementations are addressed. Moreover, implementation results in terms of functionality and thermal regulation, with a variety of traffic patterns, are presented and discussed.

Chapter 7 "Conclusions and Future Work". This chapter draws major conclusions and gives suggestions for future extensions of the works in the thesis.

2

BACKGROUND AND LITERATURE REVIEW

2.1 INTRODUCTION

With the shrinking of feature size, both the scale of integration and complexity of very large scale integration (VLSI) systems that can be placed in a chip are increasing. This motivated the development of multi-core and many-core systems-on-chip (SoCs), multiprocessor systems-on-chip (MPSoCs) and chip multiprocessors (CMPs) to deliver new levels of performance. However, buses, were traditionally the mainstay of on-chip interconnection, but these cannot keep up with the communication demands of multi-core systems. Even thought bus-based architectures have evolved from a single-shared bus to multiple bridged buses, such as in the AMBA multi-layer [177], they have remained non-scalable for large number of cores making on-chip communication a performance bottleneck.

To tackle the challenges of on-chip interconnection complexity, *Networks-on-Chip (NoCs)* have been proposed as a power efficient, modular, reliable and scalable on-chip communication paradigm [63, 23]. Since their emergence, packet-switched NoCs have became a common solution to overcome the limitations of point-to-point and bus-based on-chip communication architectures.

This chapter reviews the major concepts associated with on-chip networks, including architectures, topologies, routing, switching and flow control, in addition to emerging technologies in NoCs research such as three-dimensional networks-on-chip (3D NoCs). Moreover, recent advances in Network-on-Chip (NoC) research by the leading academic and industrial communities are surveyed.

2.2 BACKGROUND

2.2.1 *Networks-on-Chip*

NoCs have emerged as a new communication platform for connecting intellectual property (IP) cores in the same chip to form a system-on-chip (SoC) or chip-multiprocessor (CMP). On-chip communication is achieved through packet-based messaging, and routers are used to relay packets among the interconnected components. On-chip networks share many design challenges with networks at other scales, such as on-board networks of processors or computer networks. However, NoCs connect components on the same chip. This imposes tighter power and area constraits on their design.

Since the introduction of NoCs over a decade ago, rapid advances in both research [130, 145, 23, 86, 112, 110] and industry [193, 181, 183, 164] have occurred. This section briefly introduces various aspects of on-chip networks, including architectures, topologies, routing, switching, flow control and 3D NoCs.

2.2.2 *NoC Topology*

The morphological structure or connection pattern of network nodes that are connected by a set of channels is called network topology. Network topology is chosen based on many factors, such as scalability, cost and performance. Fig. 2.1 illustrates a set of the most popular on-chip network topologies that are used in research and industry [59, 66]. The simplest topology is the shared bus (Fig. 2.1a) where all IP cores share a common link and, thus, compete for exclusive access to this link. For systems with high communication requirements and large numbers of IPs, buses cannot scale efficiently and can be a performance bottleneck.

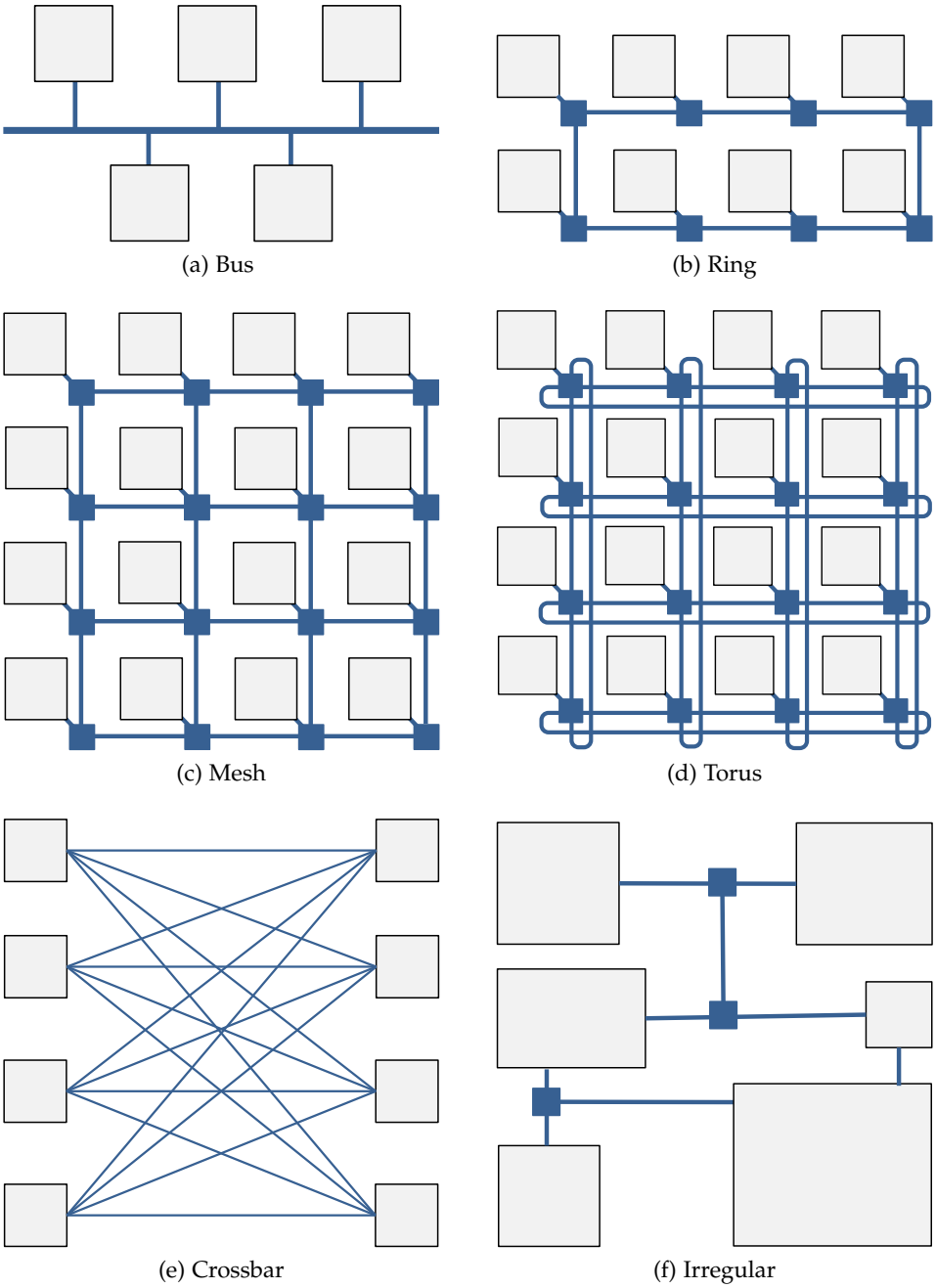
The bus can be slightly modified to create a ring topology (Fig. 2.1b) to achieve better performance. However, the ring can become saturated at a low injection rate and it is still not efficiently scalable. The crossbar topology (Fig. 2.1e) provides full connectivity which enable one-hop distance between any two IPs. However, the crossbar is poorly scalable since the number of links required increases exponentially with the number of nodes. Mesh and torus topologies provide much better performance and they are more scalable than other topologies. Thus, many commercial implementations of NoC-based CMPs and SoCs have adopted the mesh [181, 193, 164] and torus [59, 66]. On the other hand, some SoCs adopt irregular or application-specific topologies (Fig. 2.1f) that are tailored to the requirements of the target application [138, 167].

2.2.3 *NoC Components*

The architecture of the NoC and its components can vary considerably between one system and another depending on the requirements of the design. However, the description of NoC components here assumes a generic design. The generic NoC architecture consists of routers, links and network interfaces (NIs) which connect IP cores to the network, as shown in Fig 2.2. A brief description of each of these components is given bellow.

2.2.3.1 *Router*

The router is a major part of a NoC. Its function is to relay data packets in their journey from source to destination. Router architectures can



(a) Bus (b) Ring

(c) Mesh (d) Torus

(e) Crossbar (f) Irregular

Figure 2.1: Examples of common on-chip network topologies.

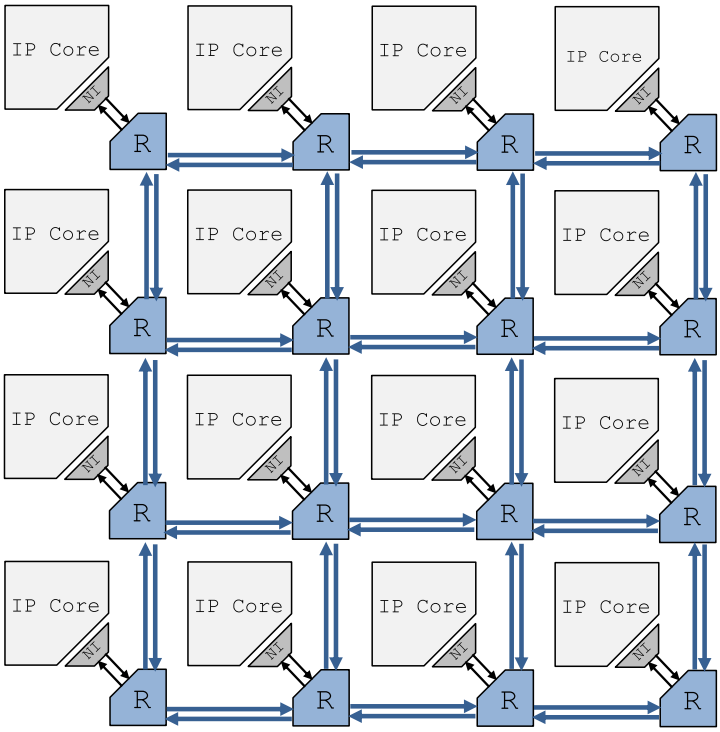


Figure 2.2: Illustration of components in a 2D mesh NoC. NI: Network Interface, R: Router.

vary significantly from one design to another as routers must be designed to meet area and power constraints, as well as, the performance requirements of the target on-chip system. These requirements and constraints vary from one on-chip system to the other and the router design varies accordingly.

A generic router architecture is shown in Fig. 2.3. The number of input/output channels, n, in the router is determined by network's topology. Each input (and in some implementations output) channel has a First-In-First-Out (FIFO) buffer that is used to store data in transit. The crossbar switch connects every input channels to output channels and, typically, provides full connectivity between input and output channels. The routing and arbitration component implements the routing algorithm in order to direct incoming messages to an output link and set the switch according to the routing decision. If multiple input channels request access to the same output channel, the arbitration circuit provides a decision.

Other router implementations may include virtual channels (VCs) and/or pipelining [60, 109] to improve router performance and to accommodate to the requirements of the system. However, such implementations can have significant overheads in terms of both area and power.

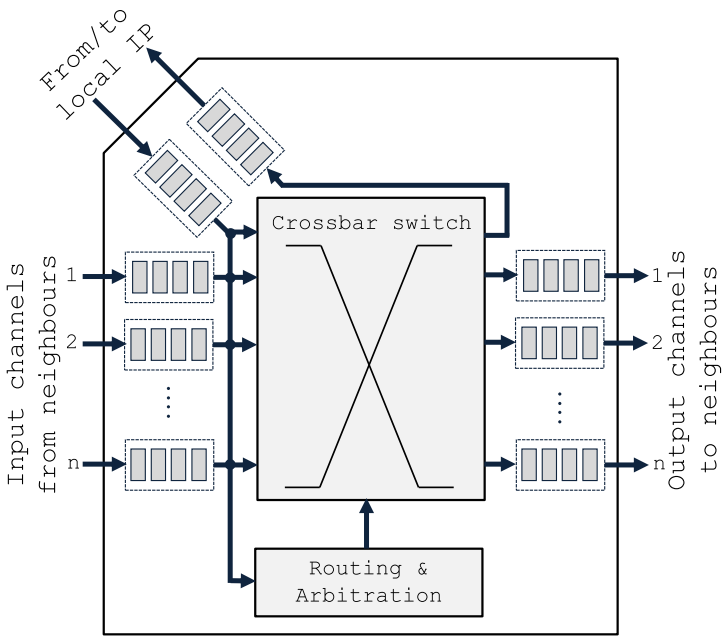


Figure 2.3: Illustration of NoC router microarchitecture [66].

2.2.3.2 *Link*

Links are metal-wire interconnects that are used to connect routers in the NoC. Typically each router has a link for each channel. Due to their length, which can cause significant signal deterioration, NoC link wires usually have repeaters (or buffers) to provide the required wire performance for reliable data transmission [116].

Due to aggressive technology scaling, on-chip links, in general, and NoC links in particular, are prone to various types of noise and performance deterioration. RC delay increases with technology scaling due to higher wire resistance and parasitic capacitance [102]. In addition, due to higher clock frequency, tighter timing constraints are imposed on link design to avoid timing violations. These factors lead NoC links to be sensitive to various types of noise. In particular, power supply and thermal noise can increase wire RC delay and repeater delay, causing adverse impacts on link performance and reliability [22, 43, 35, 116].

Besides traditional wire interconnects, other emerging technologies for on-chip interconnection have also been proposed for NoCs. Among these are radio frequency (RF) [37], optical [133] and surface-wave [113] interconnects. Despite advantages over wire interconnects, these emerging technologies are subject to various challenges and a considerable amount of ongoing research is addressing these challenges [37, 133, 113].

2.2.3.3 *Network Interface*

A network interface (NI) connects the IP core to the router and controls the sending and receiving of data packets to and from it. A packet is usually split into multiple *flits* (flow control units). The NI packetizes the data sent from the IP core to the network and adds control header data to the sent packets. The NI also depacketizes the received data and interprets the header control data in order to assemble the received message [66].

2.2.3.4 *Intellectual Property Core*

IP cores are a reusable units of logic that are the property of a particular party. These can be either open source units or proprietary units that are copyrighted to a commercial vendor. IP cores can be assembled in application-specific integrated circuit (ASIC) or field-programmable gate array (FPGA) designs. The IP can be a processing element (PE), memory, digital signal processing (DSP) element, or interfaces, such as synchronizers. IP and NoC designers must comply with standard protocols, such as the Open Core Protocol (OCP), in order to enable plug-and-play and core reusability and to reduce design effort and the time-to-market.

2.2.4 *Flow Control*

Flow control determines the way NoC resources, such as buffers and links, are allocated. Efficient flow control protocols must reduce the delay of data messages in order to increase data bandwidth. This is achieved by the effective allocation of resources to data messages. On the other hand, complex flow control implementations require complex router architectures and can incur high wiring and data overheads.

Flow control protocols are classified according to the granularity at which data is allocated. Thus, flow control can take place at message, packet or flit (flow control unit) level, as illustrated in Fig. 2.4.

2.2.4.1 *Message-Level Flow Control*

The coarsest granularity is message-based flow control. This is based on circuit switching, where buffers and links are allocated for the entire message. A setup message, or probe, is sent to reserve all the required network resources from the message source to the destination before transmission starts. The actual message data is then transmitted. After the end of transmission, the path resources are released to enable other messages to use them. A major advantage of message flow control is that it does not incur higher data overheads for packetization and depacketization. However, due to inefficient resource sharing, message-level flow control offers poor data bandwidth [109].

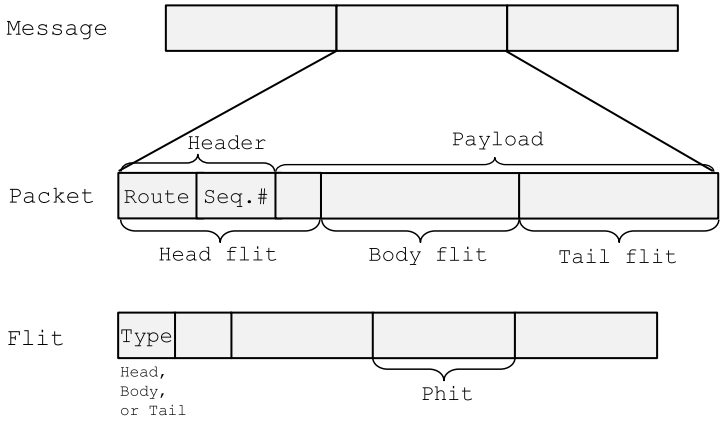


Figure 2.4: Illustration of NoC flow control units of different granularities [109].

2.2.4.2 *Packet-level Flow Control*

In packet-based flow control, the message is first broken down into packets. This enables interleaving packets belonging to different messages to use the links, thus improving links utilization. Each packet has routing information stored in the packet header and can be treated independently by the network. Also, the packet carries a sequence number to enable the packets to be assembled at the destination in order to retrieve the original message (see Fig. 2.4).

If each node stores the full packet before it is transmitted, the flow control is called *store and forward*. In this case, the routers need to provide buffering capacity to store the entire packet. Conversely, in *virtual cut-through* flow control, the node allows the packet to proceed to its destination before the entire packet is received. This reduces the delay experienced by the packets at each hop. However, data bandwidth and buffer storage are still assigned in packet-size units [109, 66].

2.2.4.3 *Flit-level Flow Control*

Flit-level flow control, or *wormhole* flow control, reduces the buffering needs of packet-level flow control by allocating buffers and links to flits instead of packets. Flits may also be broken down into *Phits* (physical units) that are equal in size to channel width. Low buffering enables the tighter area and power constraints of NoC routers to be met.

Similar to virtual cut-through flow control, wormhole flow control allows packets to proceed to the next hop before the entire packet has been received. However, the units in wormhole flow control are flits instead of packets. Thus, bandwidth and storage are reserved for flits rather than packets. The header flit is sent first and reserves the buffers and links on its way. The header flit is followed by the body flits of the same packet. The body flits use the resources previously

reserved by the header flit. Finally, the tail flit releases the resources reserved for the packet to allow them to be used by other packets. The resources reserved by the header flit of a packet cannot be used by flits of other packets until they are released by the tail flit of the reserving packet [66, 109].

Wormhole flow control reduces the buffering requirements in the router compared to packet-based techniques. Thus, it is a very popular technique due to the tight area and power constraints of NoCs. However, several links and buffers may be reserved for long packets, causing long paths to be held. This leads long paths to be prone to blockage. Moreover, the NoC is more likely to experience deadlocks when cyclic dependency occurs during resource allocation [142, 121].

2.2.4.4 *Virtual Channels*

All the above techniques are prone to head-of-line blocking if there is a single queue for each input channel. Head-of-line blocking occurs when the packet at the head of the queue is blocked which stalls all packets behind it. Virtual channels (VCs) have been proposed to overcome this problem, where multiple queues, or virtual channels, are associated with each physical channel. Virtual channels arbitrate access to the physical channel on a cycle-by-cycle basis [60]. When ahead-of-line blocking occurs, the stalled packets are allowed to traverse the link through an alternative virtual channel. Thus VCs improve physical link utilization and increase network throughput. VCs can also be used to avoid deadlocks avoidance and for quality of service (QoS) prioritization [190]. However, VCs may incur high area and power overheads due to multiple buffer storage requirements and the associated arbitration and control circuits needed.

2.2.5 *Routing Algorithms*

The routing algorithm determines the path that a packet must follow through the network nodes to reach its destination. Routing algorithms are designed with consideration given to the network topology. A good routing algorithm minimizes congestion, hotspot formation and packet delay, and increases network throughput by distributing the network load evenly among the paths offered by the network topology.

Many routing algorithms have been proposed for on-chip networks. The simplest and most popular is dimension-ordered routing (DOR). DOR is deterministic routing which means that the packets always travel between a source and destination by the same path. Fig. 2.5 illustrates XY routing, as an example of DOR, in two-dimensional (2D) meshes. In XY routing, packets are sent along the X dimension first, then along the Y dimension towards their destinations.

Another group of routing protocols is *oblivious* routing. Here the packets use different paths between a source and a destination. How-

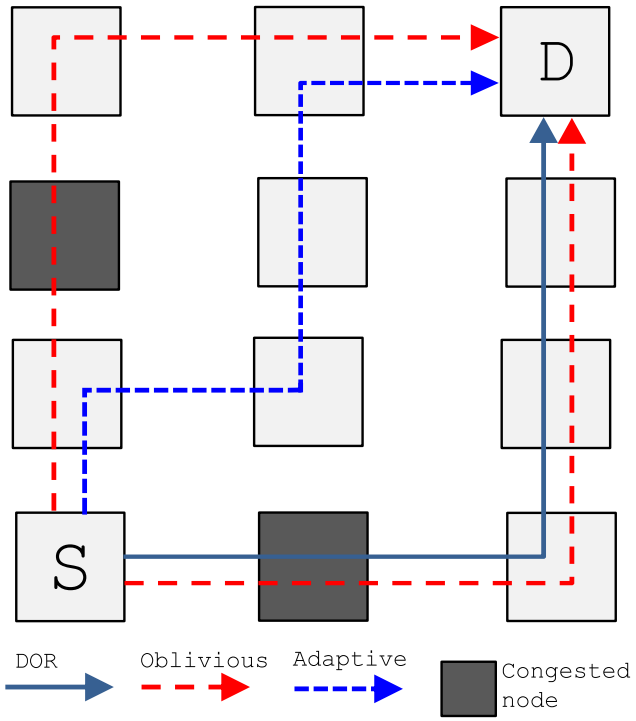


Figure 2.5: Possible paths for various types of routing algorithms for a 2D mesh, DOR show the path for X-Y routing while Oblivious alternates between X-Y and Y-X paths between a source, S, and a destination, D. For Adaptive routing an example of a possible path that avoids congestion is illustrated.

ever, the paths are chosen without regard to network status. Valiant's routing algorithm [191] is an example of oblivious routing in which load balancing is achieved by random alternation among the available paths between a source and a destination.

In both DOR and oblivious routing strategies, network paths are chosen without consideration to aspects of network status such as traffic congestion. A more complex category of routing algorithms is the *adaptive* routing. Fig. 2.5 shows an example of this, where the routing path is selected such that congested nodes are avoided in order to minimize delay and improve performance [66].

Another classification of routing algorithms can be as *minimal* or *non-minimal*. For minimal routing algorithms, the paths between a source and a destination are always the shortest possible. In non-minimal routing, paths longer than the shortest may be selected by the routing function. This may cause higher energy consumption due to higher number of hops. However, choosing paths that circumvent congested nodes can lead to lower delays despite longer paths being traversed.

2.2.5.1 *Deadlocks and Livelocks*

The main goal of every routing scheme, whether it is deterministic or adaptive, is to make sure that packets injected into the network get to their destinations eventually. A packet may not reach its destination for two main reasons: either it is involved in a *deadlock*, or a *livelock*.

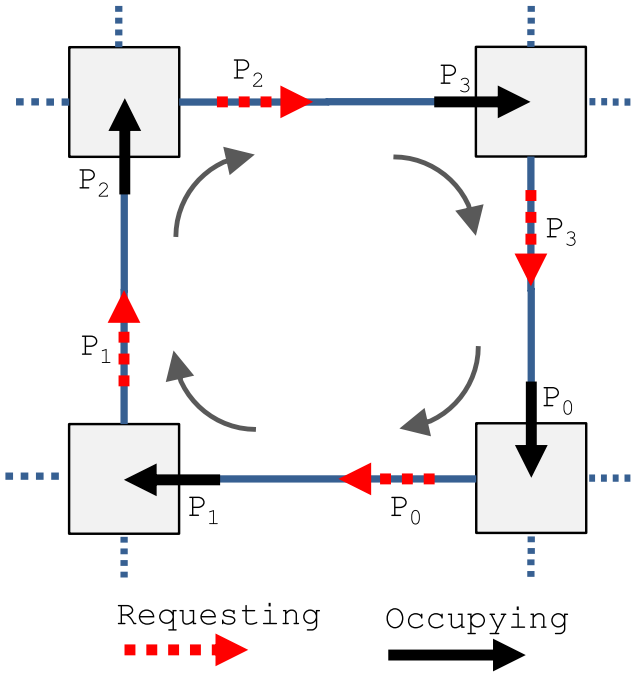


Figure 2.6: An example of deadlock scenario in which packets are forming a dependency cycle and cannot progress forward because they request channels that are occupied by other packets.

Deadlocks are of concern for any adaptive routing algorithm and exist when a dependency cycle forms among the paths of multiple messages. Fig. 2.6 illustrates an example of a deadlock scenario in which the four packets, P_0, P_1, P_2 and P_3 form a dependency cycle and cannot progress forward because they request channels that are occupied by each other. There are two main strategies to deal with deadlocks: detection and recovery; and avoidance [121].

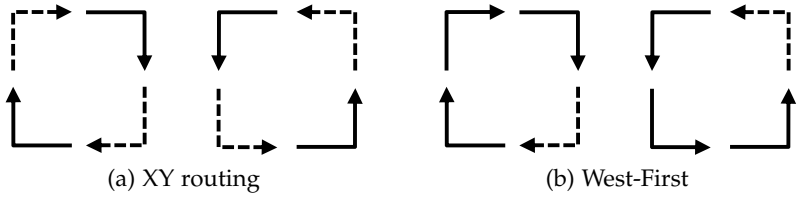
A *Livelock* means that, as time goes to infinity, the packet continues to circulate in the network without reaching its destination. Livelock can happen if non-minimal routing is used. To ensure freeness from livelock in non-minimal routing algorithms, "progressive forwarding" must be guaranteed for each packet [109, 83].

2.2.5.2 *Fully-Adaptive and Partially-Adaptive Routing*

Fully-adaptive routing means that there is no restriction on packet direction. This provides flexibility in packet manoeuvering in response to network status. However, fully-adaptive routing algorithms are

prone to deadlocks due to the possibility of cyclic channel dependency. Several techniques can be used to guarantee the deadlock freeness of adaptive routing algorithm. For example, deadlock detection and recovery techniques are used to detect deadlocks when they occur in order to enable the system to recover from them [121, 13]. Also, VCs can be used used to recover from deadlocks since they provide alternative paths for the deadlocked packets [190, 60]. However, implementing these techniques may lead to high overheads. Thus, deadlock avoidance using partially-adaptive routing or *Turn Model* techniques is preferable in many designs.

Turn model or partially-adaptive, routing guarantees deadlock-freeness in a 2D mesh topology with wormhole flow control by prohibiting just enough turns to break the possible cycles and to prohibit cyclic channel dependencies that may occur in a 2D mesh [79].



(a) XY routing (b) West-First

(c) North-Last (d) Negative-First

Figure 2.7: Allowable and prohibited turns in XY DOR and turn model routing algorithms in a 2D mesh. Dashed lines indicate prohibited turns and solid lines indicate allowable turns.

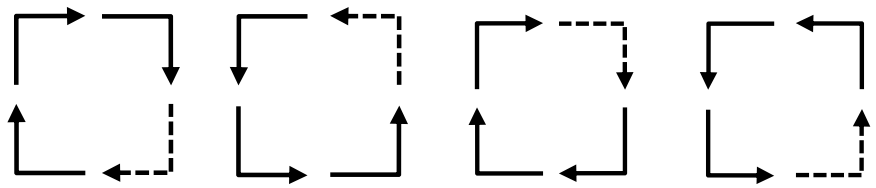
Fig. 2.7 illustrates allowable and prohibited turns in XY and turn model routing algorithms. It should be noted that, among the eight possible turns in the 2D mesh, only four are allowed in XY DOR routing (Fig. 2.7a). Although this ensures deadlock freeness, it results in deterministic routing in which the packet has only a single possible path to follow from source to destination, resulting in no path diversity. Thus, XY routing is deadlock-free but does not allow adaptiveness.

However, allowing more than four turns can still guarantee deadlock freeness. In a 2D mesh, there are two possible cycles, thus, only two turns need to be prevented to ensure deadlock freeness, one turn for each cycle. Following this concept, several turn model routing algorithms have been proposed [79]. Fig. 2.7b and 2.7c show the allowed turns in *West-First*, *North-Last* routing algorithms, respectively. In West-First no turns are allowed to the West direction while for North-Last no turns are allowed from the north direction. The *Negative-*

First (Fig. 2.7d) prohibits any turns from a positive X/Y direction to a negative Y/X direction [79].

The above turn model routing algorithms exhibit different degrees of adaptiveness for different directions. For example in West-First, if the packet is heading west, there is no adaptiveness and the path is deterministic, but when the packet is heading east the algorithm is fully adaptive. Thus, although they are deadlock-free, adaptiveness is unbalanced in the turn model routing.

The *Odd-Even* turn model provides more balanced adaptiveness by restricting turns in odd columns that are different from turns restricted in even columns (Fig. 2.8). As a result, the degree of adaptiveness provided by this model is higher [44].



(a) Odd column (b) Even column

Figure 2.8: Allowable and prohibited turns in odd-even turn model routing algorithm for odd and even columns in a 2D mesh. Dashed lines indicate prohibited turns and solid lines indicate allowable turns.

2.2.5.3 *Selection Function*

Fully-adaptive and partially-adaptive routing algorithms can give multiple candidate routing decisions (directions). Thus, a *selection function* is needed to choose among the candidate routing decisions and to select one of them based on a selection criterion. The most popular selection criterion is buffer-level (or minimum congestion) selection which gives priority for directions to the least congested nodes [66]. Other selection policies have been proposed in literature, such as maximum flexibility which tries to maximize the routing choices as the packet progresses to its destination [61]. Also, selection functions may consider the congestion in more than one hop ahead in the path, as in Neighbours-on-Path (NoP) [78].

2.2.6 *NoCs in Three Dimensional ICs*

Semiconductor manufacturing processes are approaching the physical limits. This brought attention to three-dimensional (3D) VLSI design, which could bring tremendous advantages, such as shorter global interconnects, less delay, better scalability, heterogeneous integration, lower cost and smaller form factors. Using NoCs with 3D multi-core

systems could enable the integration of unprecedented numbers of cores in future 3D CMPs.

2.2.6.1 *Three-Dimensional Integration*

Three-dimensional integration as a promising design approach that reduces system size and enables heterogeneous IC integration. The conventional 3D integration method is package-level stacking, which has been used in industry for a long time, where packages are stacked on top of each other and connected externally using wire bonding [32, 124]. Although package-level integration can achieve a significantly smaller form factor, it does not reduce wiring costs or delay. This has led attention to be directed towards die-level 3D integration, in which 2D dies are stacked vertically and connected using through-silicon vias (TSVs) [24, 124, 178, 166].

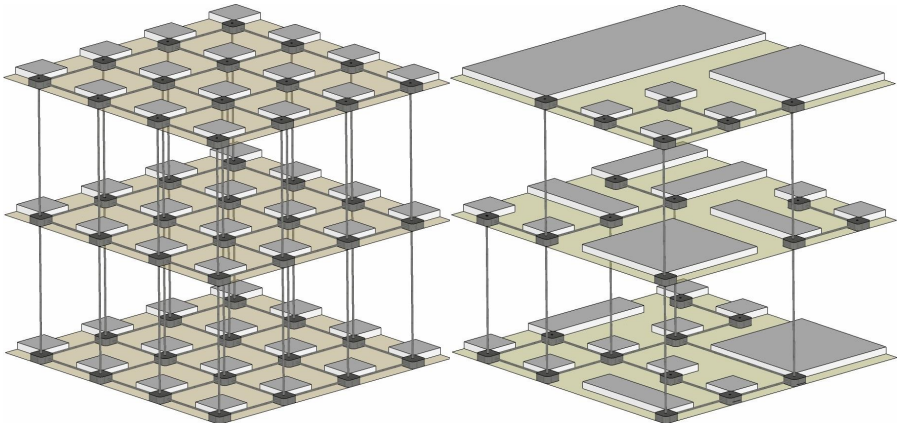
Three-dimensional integration enables the integration of heterogeneous functional blocks in a 3D SoC . These functional blocks can be implemented with different technologies, such as dense memory, optoelectronics, and high performance analogue. This is a major advantage of 3D VLSI. Other advantages of 3D include smaller form factors and shorter interconnect delay. For instance, placing the memory in a layer above the processing layer enables high memory bandwidth and lower power consumption for memory access [198, 32]. This is because TSVs are very short compared to planar interconnects, enabling high data transfer rates due to faster signal propagation, lower attenuation and reduced noise. In addition, 3D integration achieves significantly lower power consumption [32].

However, 3D integration faces many challenges, of which the thermal challenge is most prominent, leading to difficult design of chip's packaging and cooling [86, 126, 122, 8, 201]. Moreover, integrating different technologies can be difficult due to increased process complexity, higher defect rates, larger die sizes and higher cost [24]. These problems are open for research and, despite the challenges, 3D integration is considered a very promising direction for future VLSI technology.

2.2.6.2 *Three Dimensional NoCs*

The benefits of both 3D integration and NoCs are combined in 3D NoCs [204, 12, 167]. Shorter interconnects, smaller form factors and reduced delay leads to a significantly improved performance in 3D SoCs and 3D CMPs compared to 2D ones [71, 200]. Moreover, 3D NoCs exhibit higher path diversity than 2D NoCs, which translates into higher performance with adaptive routing algorithms [56, 21, 64].

Among the various 3D NoC topologies proposed in the literature, 3D mesh NoCs are the most popular due to their grid-based structure which facilitates the matching to a 2D layer layout [64, 71]. 3D mesh



(a) Homogeneous 3D NoC. (b) Heterogeneous 3D NoC.

Figure 2.9: Homogeneous and heterogeneous 3D mesh NoC geometries that uses TSVs as vertical interconnects for the stacked layers.

NoCs can be homogeneous or heterogeneous, as illustrated in Fig. 2.9. However, despite their advantages, the footprint of the TSVs is augmented significantly as the number of vertically connected IP cores increases, which puts a limit on the number of vertical links that can be embedded in 3D NoCs [153, 167]. Moreover, the thermal impact of the workload is exacerbated due to higher power density and longer heat propagation path to the heat sink [38, 209, 8]. Thermal optimization in 3D NoCs is considered in Chapters 5 and 6 of this thesis.

2.3 LITERATURE REVIEW

This section reviews NoC research conducted by various academic and industrial groups. This section is by no means a complete survey of NoC research, it's rather a review of related works that help in illustrating the contributions of this thesis. Comprehensive surveys can be found in the papers by T. Bjerregaard and S. Mahadevan [27], R. Marculescu et al [130] and R. Marculescu and P. Bogdan [131]. Reviews of specific work related to the contributions of this thesis are given in the appropriate chapters.

2.3.1 *Evolution of NoCs*

Early works that proposed the use of NoCs as on-chip communication fabric are those by Dally and Towels [63], Benini and De-Micheli [23] and Kumar et al [119]. Dally and Towels proposed using on-chip interconnection networks to replace ad-hoc global wiring and provide better support for modular design [63]. The authors proposed a packet-switched network to connect a tile-based architecture, and estimated that a 6.6% hardware overhead would be required to implement the

proposed communication fabric, in addition to a portion of the upper two wiring layers [63].

Benini and De-Micheli [23] proposed using a stacked layer approach for on-chip communication design, which adopts the existing stacked protocol paradigm that had long been used in local high-performance networks. The authors pointed out that existing and well-studied protocols, technologies and tools for computer networks represent a good candidate for future large-scale SoCs. This paradigm has many advantages, such as energy minimization, bandwidth control and standardization. Moreover, the proposed framework could provide application-specific on-chip network optimization using the layered design approach, meanwhile, flexibility would be provided by reconfigurable protocols to facilitate the plug-and-play use of components.

Kumar et al [119] proposed an architecture and design methodology for NoCs based on a mesh network that uses a five-channel switch and packet-switching. The authors also proposed a two-phase NoC design methodology based on layered inter-resource communication protocols. The first phase determines the network architecture and the second maps the application to this architecture [119].

Up to the present day, a large NoC research community has developed that investigates a wide range of topics, including routing, flow control and application (or IP) mapping in addition to traffic and power simulation and modelling [130, 145, 27].

On the other hand, various NoC implementations for research and industrial purposes have been published. These NoC designs adopt different topologies. Examples are the SPIN [9], which uses the fat-tree topology, and ×PIPES [106], which uses the torus topology. Furthermore, NoC topology can be application-specific, and several techniques have been proposed to design topologies that meet the communication requirements of an application while minimizing the use of resources [167, 67, 179]. However, the mesh topology is the most popular for homogeneous architectures and has been chosen for many NoC designs and implementations [164, 193, 181, 119].

2.3.2 *Application Mapping*

One of the most important problems that has been considered by the NoC research community is the mapping of an application to the NoC architecture. Considering the topology of the NoC, the application is broken down into a number of concurrent tasks that are assigned to a set of IP cores. Application mapping places the set of cores onto the processing elements (PEs) with the aim to optimize the metrics of interest. In other words, application mapping is used to decide how the IP cores are attached to the routers in the network. This problem is a quadratic assignment problem which is is known to be NP-hard [92].

Mapping has a considerable impact on many factors such as power consumption, and application performance [130]. Performance and cost metrics are used to determine the complexity of the mapping problem and its solution. Different strategies for application mapping in NoCs have been proposed [162]. Examples are minimizing energy [92, 137, 77, 107, 146], minimizing temperature [100, 8, 86], and maximizing path diversity [7, 186]. Table 2.1 summarizes and compares various types of NoC application mapping, including the mapping strategy proposed in Chapter 4.

Application mapping in NoCs was first addressed by Hu and Marculescu [92] who defined the metric of bit energy (E_{bit}) and used a branch and bound algorithm to map a set of IP cores to a regular tile-based NoC architecture with the objective of minimizing total energy consumption. The mapping is constrained by the bandwidth capacity of the links in order to ensure application performance [92]. Murali and De Micheli [137], presented a mapping algorithm which minimizes energy consumption and supports traffic splitting among different paths to reduce the bandwidth requirements of the links. The authors subsequently [94] proposed heuristics for energy- and performance-aware mapping followed by generation of deadlock-free routing. Increasing power density leads to thermal challenges and higher cooling costs, and thus a thermally balanced design becomes necessary [86, 8]. In [8], the authors proposed a genetic algorithm-based thermal-aware application mapping for 3D NoCs that also aims to minimize the communication costs.

Other researchers have also considered *multi-objective* application mapping in NoCs [18, 19, 186, 7]. A multi-objective pareto mapping to optimize both performance and energy consumption has been proposed in [18]. Mapping with two objectives, minimizing energy and maximizing reliability, is considered in [7, 19]. Reliability in these works is defined in terms of the path diversity that the mapping can provide for communication between cores in order to improve fault tolerance and maximize the probability of source-destination connectivity in the presence of faults.

In *irregular* architectures, where PEs are different in size, mapping must take into account that the cost of communication is not only determined by the hop count, but also the physical distance between the communicating cores. This problem was taken into consideration in [107] where the task graph and network were modelled as interconnection matrices and the problem was formulated as mixed integer quadratic programming (MIQP), which is solved using the two heuristics of successive relaxation and genetic algorithms. For application-specific irregular NoC architectures, researchers have also proposed integrating the floorplanning in the mapping algorithm to achieve an optimal floorplan in addition to mapping [138, 136, 167]. In [136], an iterative mapping-physical planning algorithm for automat-

ing NoC design has been proposed. Mapping, floorplan and topology are optimized using a *robust tabu search* to minimize NoC resources, subject to the QoS of the application being guaranteed.

Mapping in NoCs is a quadratic assignment problem whose solution may require unacceptably long computation times. Thus, some researchers have considered reducing this execution time by task graph clustering [125, 187]. For example, a simulated annealing (SA) based mapping has been proposed, where a clustering technique is integrated with simulated annealing to reduce optimization time [125].

Application task mapping can be *static* taking place during design time, or *dynamic* performed at runtime. Dynamic application mapping means that the assignment of tasks to NoC cores can change during the execution time of the application [146, 46]. These techniques cannot tolerate the long execution time of evolutionary optimization algorithms, and thus they are mainly heuristic-based. For example, a type of dynamic application mapping is proposed [146], where the mapping works as an interface between the application model layer and the architectural model layer. The mapper uses information from both layers to run a heuristic that tries to minimize communication energy at runtime.

Chapter 4 of this thesis proposes a new mapping strategy aiming to minimize power density and power supply voltage variations throughout activity balancing across the whole chip by employing a force-based optimization metric.

2.3.3 *Router Design and Routing Algorithms*

Several studies have aimed to provide efficient NoC routing algorithms and/or high-performance router architectures to implement these algorithms [150, 59, 66, 142]. These works aim to provide deadlock- and livelock- free, high-performance and reliable routing algorithms and router architectures.

On top of the DOR and the turn model routing algorithms discussed in Section 2.2.5, several adaptive routing strategies have been proposed [150, 128, 95, 202, 144, 149]. For instance, different routing schemes can exhibit different behaviour depending on the status of the network. Thus, many researchers have proposed switching between different routing strategies at runtime to combine their advantages. For example adaptive routing can outperform deterministic routing in some types of traffic patterns, while the converse is true for others. Motivated by this fact, DyAD is proposed as a routing algorithm that switches between deterministic XY routing and adaptive Odd-Even routing [95]. The switching between the two routing strategies takes place depending on the congestion status of the network. The authors also designed a prototype of a DyAD-based router which was shown to have 7% overhead compared to purely adaptive routers. Another recent work

[73] proposed the Abacus Turn Model (AbTM) as a reconfigurable routing algorithms to cope with the non-even adaptiveness of the turn model on one hand, and the dynamics of traffic which can lead to hot-spots being moved at runtime, on the other hand.

On the other hand, the characteristics of the application can be taken into consideration when designing routing algorithms in order to maximize communication adaptivity. For example, a methodology for designing application-specific deadlock-free adaptive routing algorithms for NoCs has been proposed [149], where the application is modelled as a set of concurrent tasks and the algorithm exploits the application-specific information related to communicating and non-communicating cores in order to maximize routing adaptivity and improve performance. This methodology is topology-agnostic, and thus, routing is implemented using a routing table. Another work proposes a distributed dynamic optimization architecture to update routing tables with the routing decisions of least congested paths [128].

Routing algorithm implementation must be efficient and consider the minimization of area and power costs. In general, routing algorithm for regular topologies such as mesh can be implemented using routing logic. However, for irregular topologies, or topology-agnostic router designs, table-based routing is a more convenient choice. Table-based routing can be distributed, which means that routing information is stored at each hop [59, 66]. Also, the routing table can be stored at the source, or source routing, which is common in deterministic routing and has the advantage of being simple and fast [59]. Distributed table-based routing, on the other hand, is more suitable for adaptive routing and costs less in terms of storage [59]. A distributed table-based design is adopted in the implementation of the thermal-adaptive routing strategy presented in Chapters 5 and 6 of the thesis.

The emerging 3D NoCs routing algorithms and router architectures can reuse the existing 2D adaptive routing. However, 3D NoCs routing must provide rules for inter-layer (vertical) packet communication [56, 160, 205]. A scheme for routing in irregular 3D NoCs has been proposed [160], where 2D routing is used for intra-layer routing, while for inter-layer a "gossiping" technique is used for knowing the 3D nodes (nodes with vertical links). In another work [71], the authors propose using a hybrid approach that reduces 3D router complexity. This approach uses a simple bus for inter-layer communication to exploit the short vertical interconnects which can provide significantly lower latency compared to horizontal ones. In recent work, a methodology for extending the 2D turn model routing algorithms to 3D has been proposed [56]. This provides plane-balanced adaptiveness by applying different turn-model rules in different layers.

Table 2.1: Comparison of various NoC application mapping works.

Author(s)	Target architecture	Mapping time	Optimization objective	Major contribution	Optimization algorithm
Jingcao Hu and R. Marculescu [92]	2D mesh NoC	Static	Energy	First formulation of the energy-aware mapping in NoCs	Branch and bound
S. Murali and G. De Micheli [137]	2D mesh NoC	Static	Energy	Supports traffic splitting to comply with bandwidth constraints	Heuristic-based
Z. Lu et al [125]	2D mesh NoC	Static	Energy	Proposes a cluster-based mapping to reduce optimization time	Simulated annealing
S. Tosun [187]	2D mesh NoC	Static	Energy	Clustering of both architecture and application tasks to reduce solution space	Integer linear programming
G. Ascia et al [18]	2D mesh NoC	Static	Multi-objective	Multi-objective Pareto mapping to optimize both performance and energy consumption using evolutionary algorithms	Genetic algorithms
Ababei, C. [7]	2D mesh NoC	Static	Multi-objective	Multi-objective Pareto mapping to optimize both reliability and energy consumption	Branch and bound
W. Jang and D.Pan[107]	Irregular 2D NoC	Static	Energy	Considers irregular NoC and models both the task graph and the network as interconnection matrices	Genetic algorithms

... to be continued

Table 2.1 (continued): Comparison of various NoC application mapping works.

Author(s)	Target architecture	Mapping time	Optimization objective	Major contribution	Optimization algorithm
S. Murali, L. Benini and G. De Micheli[136]	Irregular 2D NoC	Static	Topology, Floorplan and Energy	Integrated mapping, floorplannig and topology synthesis	Robust tabu search
C. Addo-Quaye [8]	3D mesh NoC	Static	Temperature	Addresses thermal minimization in 3D NoCs	Genetic algorithms
Chapter 4	2D mesh NoC	Static	Minimizing power supply noise (PSN)	Proposing a new metric, activity density, and a new force-based mapping objective to minimize power density	Simulated annealing
L. Ost [146]	2D mesh NoC	Dynamic	Energy	Uses a unified model-based approach in which mapping works as interface between application layer and architectural layer	Heuristic-based
C. Chou and R. Marculescu [46]	2D mesh NoC	Dynamic	Energy and contention	Incorporates the user behaviour information in the mapping process	Heuristic-based

Routing in 3D can also be application-specific, and various techniques for allocating routing paths that are optimized for specific applications in 3D NoCs have been proposed [205, 167].

2.3.4 *NoC Architectures and Design Philosophies*

VLSI academic research groups and companies have proposed various NoC architectures with different design philosophies and objectives. Examples are the MIT RAW chip [181], Æthereal [81], Intel TeraFLOPS [193], spiNNaker [148] and CONNECT [152]. In this section a summary of the design choices made in selected NoC implementations is presented.

2.3.4.1 *×PIPES*

The ×PIPES [58] was developed by the University of Bologna, Italy and Stanford University, USA. It is a library of parametrizable and synthesizeable NoC interface, switch, and link modules that can support both homogeneous and heterogeneous architectures. The developers of ×PIPES also proposed ×PIPES compiler [106] which is a tool for instantiating a network optimized for specific applications from the library of macros (switches, interfaces and links). ×PIPES links are pipelined to decouple the data introduction rate from link delay and to make data injection delay-insensitive. A cyclic redundancy check (CRC) error control is implemented at the link-level with a go-back-N retransmission. ×PIPES uses packet switching with flit-based wormhole flow control and adopts the OCP 2.0 for interfacing with SoC cores. Static source routing is implemented in the NI by a lookup table based on the destination address. Although source routing incurs high header overheads it allows a lightweight switch implementation [58].

2.3.4.2 *MIT RAW*

Developed by MIT, the RAW [181] many-core processor consists of 16 identical programmable tiles. Each tile contains a communicational block and a computational block. The computational core encompasses an eight-stage, in-order, MIPS-style processor; a four-stage, pipelined, floating-point unit; a 32-Kbyte data cache; and a 96 Kbytes of software-managed instruction cache. The RAW chip is organized as a 2D mesh topology with a communication fabric that consists of four networks, two static and two dynamic. The static networks are used for compile-time predictable communication to provide the low-latency communication required for software circuits and other applications. The static router uses a five-stage pipeline that controls two routing crossbars, and thus two physical networks. The dynamic network, on the other hand, transports unpredictable operations such as interrupts, cache misses, and compile-time unpredictable communi-

cation (for example, messages) between tiles. The dynamic networks are wormhole DOR routed. To configure the dynamic network, the user injects a single header word that specifies the destination tile (or I/O port) and the length of the message which must not exceed 31 flits. The RAW processor supports instruction-level parallelism (ILP), and evaluation with a range of applications shows that its performance is close to that of the specialized machines for these applications [182].

2.3.4.3 *Nostrum*

Nostrum [3] is a research project at the KTH Royal Institute of Technology which aims to develop a NoC architecture for SoC in multiple application domains. Nostrum is a packet-switched regular, two dimensional mesh NoC. The Nostrum network is pseudo-synchronous, meaning that the clock frequency for the switches remains the same but phases may vary. The routing is adaptive, deflective routing which allows for adaptivity against faults and congestion. Moreover, with deflection routing, fewer (or possibly no) buffers are required, which minimizes the buffering cost of the routers. Nostrum provides both best effort (BE) QoS and guaranteed service (GT) QoS. Guaranteed service is provided by virtual circuits which are implemented using two concepts. The first, *looped containers*, are information currying packets that looped between the source and the destination; and the second, *temporally disjoint networks*, is an explicit time division multiplexing mechanism [132]. The main applications targeted by Nostrum are processor networks and multimedia [3].

2.3.4.4 *MANGO*

The MANGO [26] network was developed at the Technical University of Denmark, where MANGO stands for *message-passing asynchronous network-on-chip providing Guaranteed services over OCP interfaces*. MANGO is a clockless NoC which is designed for coarse-grained globally asynchronous locally synchronous (GALS) SoCs. It provides connectionless BE routing as well as connection-oriented GT. To achieve a simple design, the VCs are implemented as separate physical buffers such that the GT can be provided by allocating a sequence of VCs through the network. The links include delay-insensitive signal encoding, which renders global timing robust. The MANGO NI adapters are OCP-based, and can synchronize the clocked OCP interfaces to the clockless network.

2.3.4.5 *Æthereal*

Developed by Philips research laboratories, the Æthereal [81] aims to provide a complete infrastructure for developing heterogeneous NoC, with the performance levels required for real-time systems. The Æthereal NoC provides both GT and BE services. For GT, there is a

common sense of time in all the NoC routers such that the routers forward traffic based on time slot allocation. This provides guaranteed throughput, data integrity and the bounded latency required by traffic in real-time applications. Slot allocation can be conducted statically, at system initialization, or dynamically at runtime. The BE service, on the other hand, is used for timing unconstrained applications and is provided throughout the conventional wormhole flow control, where BE traffic uses non-reserved slots or any slots that are reserved and not used by GT traffic. The authors have described [80] a design flow for the instantiation of application-specific Æthereal NoCs with given application communication requirements. The design flow uses XML to input various parameters such as traffic characteristics, GT and BE requirements, and the topology is designed to ensure that the application communication requirements are guaranteed. Furthermore a high-end TV system architecture with the Æthereal NoC has been proposed [176].

2.3.4.6 *Intel Many Integrated Core (MIC) Architecture*

The Intel Many Integrated Core (MIC) architecture is a multiprocessor computer architecture based on Intel's research on many-core architectures, which includes the TeraFLOPS Research Chip project [193] and the Single-chip Cloud Computer (SCC) [164]. The TeraFLOPS prototype research chip was developed by Intel's Tera-Scale Computing Research Program [89]. Fabricated using a 65 nm eight-metal CMOS process, this experimental tile-based many-core processor chip includes 80 tiles arranged as an 8×10 2D array. Each tile has a PE connected to a 5 port router. The PE has two pipelined single-precision floating-point multiply-accumulate units (FPMAC0 and FPMAC1), 3 KB single-cycle instruction memory (IMEM), and 2KB of data memory (DMEM). The routing scheme is source-directed, the router frequency is 4 GHz and it has 5 ports, and uses wormhole-switching. The router is also equipped with a two-lane pipeline for deadlock-free routing. Each lane has a dedicated 16 FLIT queue, arbiter and flow control logic. The 2D mesh network-on-chip has been reported to achieved a bandwidth of 2 terabits/sec. This chip supports fine-grain power management and is reported to achieve a computing performance of over 1.0 TFLOPS with a power supply dissipation of 97 W at 4.27 GHz and 1.07 V supply [193].

Following the TeraFLOPS, Intel's Tera-Scale Computing Research Program introduced the Single-Chip Cloud Computer (SCC) [164]. SCC is an experimental many-core processor and was announced in 2009. It supports "scale-out" message-passing programming models that have been proven to scale to 1000s of processors in cloud data-centres [165]. The SCC chip contains 24 tiles arranged as a 4 × 6 2D mesh. Each tile has two Pentium IA-32 cores, two 256 KB private L2 caches (one for each core) and a router. Moreover, the core to core

communication is expedited by a 16 KB Message Passing Buffer (MPB). Fabricated using 45nm technology, the SCC chip contains a total of 48 cores, each capable of running a separate operating system and software stack and acts like an individual computational node that communicates with other computational nodes over a packet-switched network [165]. The 2D mesh NoC has 24 routers connecting the chip tiles. The router has 5 ports and uses a fast virtual cut-through routing protocol with eight VCs and two message classes: request and response. The DOR XY routing and route pre-computation in the previous hop to allow a quick packet forwarding and improve network utilization. SCC supports dynamic voltage and frequency scaling (DVFS), with voltage scaling allowed at the granularity of a group of four tiles and frequency scaling is allowed at single tile granularity. Moreover, hierarchical clock gating at tile level and at individual ports of the router is provided. These power management features enable all 48 cores to run at the same time over a range of 25W to 125W. The performance improvement of the SCC chip is reported to be 2.8× over its predecessor, the TeraFLPOS chip [165, 164].

Recently, following these prototypes, Intel announced the first commercial MIC architecture, the Knights Corner, which uses 22 nm and scales to more than 50 processing cores on a single chip [101].

2.3.4.7 *SpiNNaker*

SpiNNaker is a massively parallel computer architecture developed at the University of Manchester inspired by the working of the human brain [5]. SpiNNaker is a GALS system with processing nodes located in synchronous islands and surrounded by a packet-switched asynchronous NoC. Each SpiNNaker multiprocessor system-on-chip (MPSoC) has a CMP, with 20 ARM968 processors, and a 1Gbit mobile DDR SDRAM memory. The original aim of the project was the modelling of large systems of neurons in packet-based spike communication, which is computationally demanding in terms of processing power and communication [114, 74]. However, it is now employed by many applications and research studies in robotics, computer engineering and science, and neuroscience [5]. The communication among processing nodes is based on a multicast infrastructure inspired by neurobiology. A packet-switched NoC is used to emulate the densely connected neuronal systems. The packets are source-routed and only carry information about the packet source. The multicast communication is controlled by a bespoke multicast router which uses a routing table and can replicate packets, if necessary, to send the same packet to several different destinations. The SpiNNaker chip supports inter-chip communication through six bidirectional asynchronous channels allowing the connection of many SpiNNaker chips using off-chip networks with different topologies. The target of the project is to be able to simulate a neural network consisting of one billion simple neurons,

requiring a machine with over 50,000 chips and a total of one million processing core which is expected to consume nearly 100 KW of power [5].

2.3.4.8 *CONNECT*

The CONfigurable NEtwork Creation Tool (CONNECT) [152], is a flexible RTL generator for FPGA-friendly NoCs. This tool was developed at Carnegie Mellon University and the developers also provide a Bluespec SystemVerilog-based front-end NoC generation framework which is freely available online [2]. The NoC generator can produce synthesizable RTL designs of FPGA-tuned multi-node NoCs with various topologies in Verilog HDL. The CONNECT involves many design principles that are motivated by FPGA architecture. Thus, many design decisions, such as topology, link width, router pipeline depth, network buffer sizing, and flow control, are optimized for FPGA. For example, as a result of the abundance of wires in the FPGA, wide datapaths and channels between routers are used. Wider channels can also compensate for the lower bandwidth of FPGA due to its low frequency compared to ASICs [152]. Evaluation of the CONNECT in FPGAs shows significant reductions in resource costs compared to state-of-the-art NoCs designed for ASIC with comparable performance [152].

2.3.5 *NoC Models and Simulators*

A considerable amount of studies have been published on the simulation and modelling of NoC systems, and several network simulators and modelling tools are proposed. Performance evaluation has been a major concern in several works [70, 1, 105, 135]. Other proposed simulators and models also focus on power and area modelling [112, 188]. Table 2.2 summarizes and compares the main features and of various on-chip network simulators and modelling tools, including the simulators and models developed in this thesis and employed in Chapters 3 and 5.

Table 2.2: Comparison of various NoC models and simulators.

Name	Developed by	Supported Topologies	Routing Algorithms and Switching	Supported traffic patterns	Configurable NoC Parameters	Evaluated Parameters	Comments
Booksim [1]	Stanford University	Various including 2D mesh, torus, flattened, butterfly, fat tree, quad tree, etc.	- Various including: deterministic DOR, adaptive, minimal, non-minimal, etc. - Wormhole with VCs	Uniform random, bit complement, bit reversal, shuffle, transpose, tornado, neighbour and random permutation	- Number of VCs, Buffer depth, router processing delay, allocator type, flit size and NoC size	Performance	- Developed with C++. - Cycle-accurate NoC simulator. - Covers wide range of topologies and routing algorithms.
Noxim [70]	University of Catania	2D mesh	- Various including: deterministic DOR, turn model and fully-adaptive - Wormhole	Uniform random, bit reversal, shuffle, transpose, hotspot, butterfly and table-based	Buffer depth and NoC size	Performance, Power	- Cycle accurate SystemC simulator - Easy to expand and integrate with other tools - Include dynamic power simulation
Orion [112]	MIT	Unrestricted	- Fully-adaptive - Wormhole with VCs	Not applicable	Various including: buffer depth, flit size, crossbar type, link size, link length, frequency, etc.	- Router power and area	- High-level power and are model for NoCs - Include many microarchitectural choices - Technology parameters are deliverable from standard technology files

... to be continued

Table 2.2 (continued): Comparison of various NoC models and simulators.

Name	Maintained by	Supported Topologies	Routing Algorithms and Switching	Supported traffic patterns	Configurable NoC Parameters	Evaluated Parameters	Comments
Netmaker [135]	Cambridge University	Unrestricted	- Deterministic XY - Wormhole with VCs	Uniform random	Buffer depth, Flit size and NoC size	Performance	- SystemVerilog-based simulator - A library of synthesizable and parameterized NoC implementations - Pipelined and speculative router architectures
Chapter 3	Newcastle University	2D mesh	- Various including: deterministic DOR, turn model and fully-adaptive - Wormhole	Uniform random, bit reversal, shuffle, transpose, hotspot, butterfly and table-based	Various including: buffer depth, flit size, crossbar type, link size, link length, Frequency, NoC size, etc. In addition to power delivery parameters and chip floorplan	Power supply noise (PSN), performance and power	- Developed based on Noxim simulator - Embeds an updated power and area models (Orion), and floorplan information - Integrated with a fast power delivery model
Chapter 5	Newcastle University	2D and 3D meshes	- Various including: deterministic DOR, turn model and fully-adaptive - Wormhole	Uniform random, bit reversal, shuffle, transpose, hotspot, butterfly and table-based	Various including: buffer depth, flit size, crossbar type, link size, link length, frequency, etc. In addition to thermal chip parameters and floorplan	Dynamic temperature distribution, performance and power	- Developed based on Noxim simulator - Embeds an updated power and area models, and floorplan information - Integrated with a microarchitectural thermal model (Hotspot [96])

... to be continued

Table 2.2 (continued): Comparison of various NoC models and simulators.

Name	Maintained by	Supported Topologies	Routing Algorithms and Switching	Supported traffic patterns	Configurable NoC Parameters	Evaluated Parameters	Comments
Nirgam [105]	University of Southampton and Malaviya National Institute	2D mesh and 2D torus	- *XY* and *odd-even* - Wormhole with VCs	Customizable constant bit rate, trace-based and bursty traffic	Buffer depth, flit size, NoC size, number of VCs and frequency	Performance	- SystemC cycle accurate simulator - Plug-in support for applications
NoCTweak [188]	University of California, Davis	2D mesh	- Various including: deterministic DOR and turn model routings - Wormhole with VCs	Customizable constant bit rate, trace-based and bursty traffic	Buffer depth, flit size, NoC size, number of VCs, technology node and frequency	Performance and power	- SystemC cycle accurate simulator - Supports synthetic and real embedded application benchmarks

3 MODELLING AND ANALYSIS OF POWER SUPPLY VARIATIONS IN NOCS

3.1 INTRODUCTION

Power supply noise (PSN) has adverse effects on digital circuit performance and reliability. It can cause signal deterioration and create soft errors. Recently, it has been reported that variations in power supply can have significant impacts on operational frequency and system power dissipation [163, 141]. Both resistive (*IR*) and inductive (ΔI) voltage drops are sources of PSN. Resistive voltage drop occurs mainly due to the resistance of power delivery wires in the power grid network and increases with the amount of current delivered through these wires. On the other hand, inductive drop is mainly due to wire inductance in the package as well as in the grid wires and is proportional to the rate of change of current.

Technology scaling exacerbates the problem of PSN for many reasons. Firstly, the thickness of the wire used in the power networks is rapidly shrinking. This substantially increases the resistance of power delivery wires. Also, demand for power delivery is rapidly increasing, and both these factors lead to higher *IR* voltage drop. Secondly, higher switching frequency increases ΔI drop. Thirdly, a lower operating voltage decreases the noise margin. Consequently, voltage drop as a percentage of supply voltage is rapidly increasing. For example, the voltage drop can be up to 30% of nominal supply voltage in 65 nm technology if the necessary precautions are not taken [11]. Mitigating PSN has become a significant challenge in ensuring the sustainability of future large-scale integration development.

The modelling and mitigation of PSN is traditionally conducted at circuit level, for example the IR analysis of power delivery and the use of decoupling capacitors for PSN reduction [39, 40]. Also, for systems with power gating capability, optimal power-gating and scheduling may be used for the mitigation of PSN [197]. However, these techniques do not include an accurate model of application activity and they may not be sufficient for accurate modelling, particularly with the power integrity challenge that is increasing with aggressive technology scaling. In addition to these techniques, high-level techniques, or techniques that consider the application's characteristics, are becoming necessary. For instance, an optimized workload assignment to the cores in multi-core systems can result in significant PSN reduction [185, 184]. However, in these techniques independent tasks are considered and

the intra-chip communication workload and task interaction is often ignored [185, 184, 85].

With the emergence of multi-core and many-core systems, dedicated and high performance on-chip communication systems are required. The NoC has been proposed as a promising infrastructure to deliver scalable and high-performance on-chip communication [63]. However, the power budget of the NoC takes up a significant portion of the overall budget NoC-based systems. For instance, the routers in the MIT-RAW CMP network consume about 40% of the tile power, and the communication network takes up to 35% of the overall system power [196]. Moreover, it has been reported that the communication power budget is about 28% of Intel's 80-tile TeraFLOPS CMP [192]. This implies that the on-chip communication workload is responsible for a considerable portion of the overall PSN. In contrast to conventional models for logics or microprocessors, this portion of PSN has interesting correlation to the temporal and spatial distributions of the traffic load. This load can be determined in the early design stages, once the application's characteristics are known. Thus, the PSN that results from on-chip communication workload can be characterized and studied at the early stages.

On the other hand, due to aggressive technology scaling, on-chip interconnection networks are exposed to various sources of noise. Apart from PSN, which is a major source of errors, process variation, crosstalk noise, thermal noise and leakage are other examples of sources of error. All of these types of noise can cause errors and contribute to performance degradation in the NoC and the overall system. Thus, error control techniques are needed for fault tolerance and to provide the QoS required by the target application [194, 211]. More importantly, accurate estimations of error and fault rates from various sources are required at the stage of the design [84]. For independent noise sources, fault rates can be modelled separately and their effects added to give a general estimation of fault tolerance metrics.

In this chapter, a tool for analysing PSN in networks-on-chip is presented. It captures the supply voltage variations caused by communication loads across the chip. The tools and models proposed in this work enable a better understanding of the trade-offs existing in the design of communication links and, in particular, allow the relationships between parameters, such as voltage or frequency, and fault rate or bit error rate (BER) to be evaluated. These relationships are crucial for the analysis of the QoS of communication fabrics in NoCs in the early stages of design. The major contributions of this chapter can be summarized as follows:

1. The development of a tool which employs an integrated model of PSN in NoCs. Detailed circuit level design parameters and application-specific on-chip communication dynamics, including traffic pattern and link bandwidth, are considered. This tool

provides a compact model which integrates NoC power and area models, an NoC simulator, on-chip link model and a power grid model.

2. A rigorous evaluation of the model accuracy is conducted. Also, the impact of power grid granularity on accuracy is analysed. This also gives an insight into the scalability of the power grid model and the trade-offs between simulation time and accuracy.

3. The model has been employed to analyse PSN in networks-on-chip. Novel observations about PSN distribution and variation due to the use of different routing algorithms and traffic patterns are found.

4. The impact of the resulting PSN on performance has been studied. Statistical timing analysis of the link delay caused by the PSN is performed. Moreover, high level fault metrics, such as the probability of timing errors and bit error rates, are also evaluated based on real-world and synthetic communication scenarios.

3.2 RELATED WORK AND BACKGROUND

NoCs are used to connect components on the same chip. The transfer of data is achieved in a way similar to that in conventional computer networks, where packet switching is used and packets are routed from the source to the destination. A packet is split into smaller data units called *flits*. The interconnected components may be general purpose microprocessors, memory blocks or control circuitry. Each component (IP) is attached to a router which is used as a gateway to connect the IP to other IPs and to route information for the overall system. The term *tile* is often used to stand for the IP core and the corresponding router.

Many tools have been developed to model NoC power and area for early-stage design space exploration [88, 167, 112]. These tools aim to help in evaluating a design at the early stages and to explore different design strategies and techniques to give an initial estimation of the significance of a specific design technique. Other researchers have focused on optimizing floor planning and topology [179, 110, 167], and application mapping [92, 137]. The majority of these efforts aim to minimize area and power and do not consider the ever-increasing problem of PSN which is directly affected by the output of these design strategies. Optimizing NoC design for PSN requires the accurate modelling of this noise in order to guide and evaluate the optimization process.

Power noise modelling requires models for both workload and the power delivery grid. The workload models determine the values and locations of the power-consuming modules in the chip, while the power grid model determines the supply voltage variations across the

power delivery grid in the presence of these workloads. This section surveys the state-of-the-art in these two areas.

3.2.1 *Power Grid Modelling and Analysis*

To analyse the power delivery grid in VLSI circuits, the grid is modelled as an RLC network, while loads are often modelled as independent current sources [28] or equivalent passive elements [120]. Determining the node voltages in an m node power grid model requires the following system of partial differential equations (PDEs) to be solved:

$$Gv(t) + Cv'(t) = i(t) \tag{3.1}$$

where $G, C \in \mathbb{R}^{m \times m}$ are matrices representing memoryless elements (resistors) and memory elements (inductors and capacitors) respectively, while, $v(t)$,$i(t)$ ($\in \mathbb{R}^m$) are vectors of voltages and imposed independent current sources at the grid nodes, respectively. In this model, independent current sources are used to represent circuit activity across the chip [28, 156].

Due to the enormous number of elements and nodes in the grid, solving the equations for the node voltages $v(t)$ using traditional circuit simulators, such as SPICE, is impractical in terms of both the memory and simulation time required. This problem has been considered by many researchers. Several solutions have been proposed to solve the power grid size problem during both simulation and modelling.

In terms of modelling, model order reduction (MOR) approaches have been used to reduce the model order before simulation. Multigrid-like [118, 82], hierarchical [207, 171], partition-based [87], and Krylov subspace-based methods [34] are examples of MOR. Other works focus on reducing simulation time, for instance in random walk-based simulation [31]. Most of these methods are based on iterative computation, and due to their high computational demand, they are difficult to use for an analysis that involves solving the model many times, in order to account for run-time application dynamics. Direct methods are instead preferable in such cases, particularly when the peak noise rather than a detailed time profile of the noise is of interest.

For example, a fast and direct model to determine peak PSN has been proposed [208]. The power grid is modelled as a distributed RLC network excited by constant voltage sources, and switching capacitors (C^{load}) are used to model on-chip circuit activity. The number of these capacitors for a particular circuit is determined by the amount of charge (or energy) delivered to the circuit during the switching time period t_s. The noise impulse is approximated with a linear ramp

which reaches its maximum at $t = t_s$, and the minimum voltage at node j in the grid, V_j^{min}, is given by:

$$V_j^{min} = \frac{1}{\lambda_j}\left(\sum_{i=1,i\neq j}^{k} x_{i,j}V_i^{min} + \frac{1}{2}\sum_{i=1,i\neq j}^{k} C_{i,j}V_{DD}\right) \tag{3.2}$$

where $\lambda_j = \sum_{i=1,i\neq j}^{k} x_{i,j} + \frac{1}{2}\sum_{i=1,i\neq j}^{k} C_{i,j} + C_j^{load}$ and $x_{i,j} = t_s^2/(6L_{i,j} + 3R_{i,j}t_s)$. $R_{i,j}$, $L_{i,j}$ and $C_{i,j}$ are the resistance, inductance and capacitance between nodes i and j in the power grid, respectively; t_s is the switching time; and C_j^{load} is the equivalent capacitance of the load at node j.

This model provides an accurate estimation of peak V_{DD} drop and a maximum error of 5% has been reported [208]; yet the simulation is significantly reduced. This seem to be an ideal candidate for developing a tool for the evaluation of real-time supply variations. However, this model assumes load-equivalent capacitances are known. In the present work, these capacitances change dynamically in real-time; thus, this power grid model can be adopted after introducing a technique to determine the switching load capacitances for the NoC system. This workload model characterizes the switching modules in the system and should capture application dynamics.

3.2.2 *Workload Modelling*

Many studies have been published on workload modelling for power grid analysis. Techniques that represent the workload with equivalent passive elements have been reported. For example, a macro model based on the effective impedance of the current consumer has been proposed [120]. Other techniques based on independent current source models are also used, where, for example, macro-models can be used to determine the waveforms of these current sources. Another work proposes a frequency domain current macro-model [28] where the input vector pairs of the circuits are partitioned according to the hamming distance, and a current macro-model is built for each distance using regression. However, such workload models assume that computation workloads can be represented as independent current sources or passive elements. Task dependencies and correlations between computational cores are ignored. More importantly, workload due to communication cannot be captured in these models.

Independent current sources or passive elements can be used as reasonable representations of workloads for simple logics or, to an extent, for microprocessors. For complex on-chip communication systems such as networks-on-chip, dynamic power consumption through the on-chip communication infrastructure leads to significant variations in power and load. Therefore, an effective model that captures the communication dynamics is required. This model must integrate the

workloads of both the router circuit and links. For the former, a router microarchitectural power model [112] can be integrated with a circuit activity simulator to characterize the router circuits workload.

For links, an on-chip link model is essential for determining on-chip communication loads. A number of models based on current [189], energy [173], and power [41, 155] for on-chip interconnects have been proposed. On-chip interconnects are modelled as capacitively and inductively coupled distributed RLC lines. An analytical model for on-chip link current based on decoupling techniques have been proposed [189]. On-chip link wires are driven by an exponential voltage source (V_S) and loaded by a capacitor (C_L). A two-port network with source and load impedances is employed to derive a closed form for wire current. The link current can then be obtained using decoupling transformation [41].

3.3 METHODOLOGY

On-chip communication traffic produces a considerable portion of the overall PSN in NoCs. In contrast to conventional models for logic or microprocessors, this portion of engendered noise or V_{DD} variation would have an interesting correlation with the spatial and temporal distributions of the traffic load. This load results from design decisions at the network-level, such as application mapping and routing path allocation (or routing algorithm).

A model of power supply noise requires a detailed consideration of both workload and power grid models. Fig. 3.1 illustrates the different components of the proposed model and the computational flow among these components. The technology and architectural parameter files, in addition to the floorplan information, are taken as inputs to the model. These files are given to the NoC power model to compute the power traces of NoC components. For the router, the well known NoC router power and area model Orion [112] is used. This model is fast, accurate and easy to integrate with other models. More importantly, it is an architecture-level model which enables the macro-modelling of the power grid workload for NoCs. Application characteristics files are fed to an NoC simulator to generate traffic information. The open-source SystemC-based NoC simulator, Noxim [70], is modified and employed here because it supports a wide range of traffic distributions and routing algorithms and due to its efficiency and ease of configuration and expansion. The fast peak noise power grid model proposed in [208] is used for power grid modelling.

Integrating all these models in an automated flow enables the computation of dynamic voltage variations in NoCs. The methods and tools presented in this chapter are applicable to a wide range of topologies, including the tree, mesh, and torus. However, for convenience it is described in the context of a regular mesh topology.

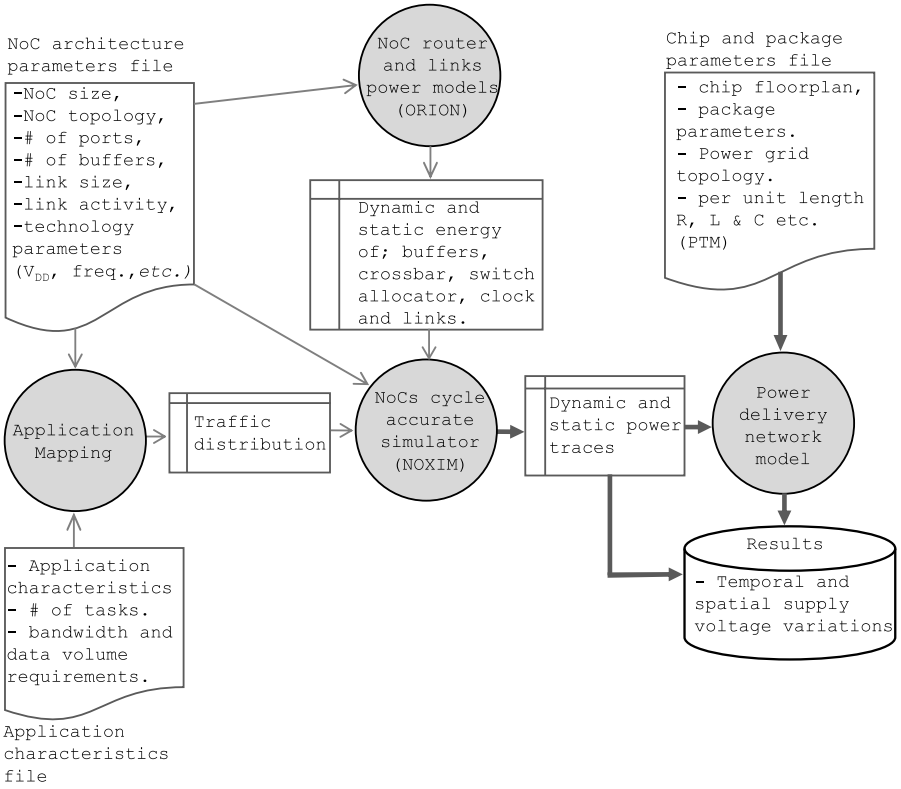


Figure 3.1: Computational flow for NoC PSN modelling.

3.3.1 *Power Noise in Networks-on-Chip*

Many design parameters can affect the spatial and temporal distribution of communication loads in NoCs. Routing algorithms, application traffic patterns, and packet injection rates are examples of these parameters. Consequently, NoC communication workloads have spatial and temporal distributions which are determined by the design of the system [10, 29, 30]. This distribution in time and space of communication loads is reflected in the spatial and temporal PSN distribution in the power delivery grid.

Fig. 3.2 shows a general overview of a network-on-chip and its power grid. The power grid is a grid of metal wires, which can be modelled as an RLC network, and may have different topologies, such as a mesh, tree or irregular topology. These grids usually span several metal layers and they are hierarchical in nature. This implies that segment length and width decreases (i.e. grid granularity increases) when moving from more global to more local power grid nodes. Some nodes in the upper layers are connected to package V_{DD} pads. Controlled collapse chip connection (C4) pads are assumed and modelled as RL segments as shown in Fig. 3.2.

A power grid analysis for the whole grid is not practical due to the huge size of the resulting model. Thus, by exploiting the hierar-

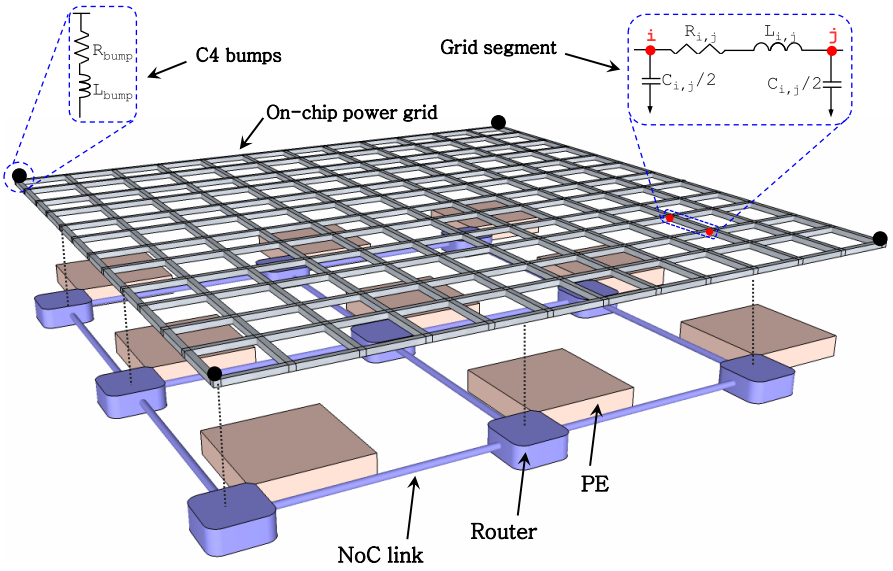


Figure 3.2: Illustration of NoC power delivery network and its RLC model. $R_{i,j}$, $L_{i,j}$ and $C_{i,j}$ are the resistance, inductance and capacitance of gird wire segment between nodes i and j. R_{bump} and L_{bump} are resistance and inductance of the C4 bumps.

chical nature of power grids, the analysis can be performed using a macro-modelling approach. A lumped model is used to characterize individual blocks in the chip and a power grid of appropriate granularity is considered [207, 85]. In this work a macro model is also considered for the routers. The router is matched with a region of the power grid which is determined by the floorplan information.

Fig. 3.3 illustrates links and routers of network-on-chip. The router's functional units and the link's equivalent circuit, including the drivers and loads, are also shown. In this work, the link wires are modelled as RLC interconnects driven by exponential voltage sources and loaded by capacitances (C_L). The NoC workload consists of the switching workloads of both router circuits and links. Router's workload is due to internal processes in the router. These processes are: receiving a flit, route computation, switch allocation and switch traversal. Link workload is attributed to the switching of the drivers of the link and the repeaters along its wires.

3.3.2 *Compartmental Modelling for Communication Fabrics*

Routers are responsible for relaying data packets in NoCs, and they are composed of a cross-bar switch, routing and arbitration logic, input channels and output channels, as shown in Fig. 3.3. For a mesh topology, there are four input/output channels for global communication (north, east, south and west) and one input/output channel for local communication. All input channels have buffers. Links connect routers and each link is driven by drivers which are part of the sending router

Table 3.1: Notation and symbols used in this chapter.

Symbol	Definition
$C_r^{total}(k)$	Total load equivalent capacitance for router r.
$C_r^{links}(k)$	Load equivalent capacitance for router r at cycle k due to link traversals for all the links in the router.
$C_{ch}(k)$	Load equivalent capacitance at cycle k due to link traversal of router channel ch.
$C_r^{circuit}(k)$	Load equivalent capacitance of router r at cycle k due to circuit activity.
n	Number of wires in the NoC data link.
R	The set of all routers in the NoC.
CH_r	The set of all channels in router r.
G_r	The set of power grid nodes responsible for delivering power to router r.
SW	Vector of size n with elements representing the wire switching direction, 0 for quiet, 1 for switching up and -1 for switching down.
Ψ	The set of microarchitectural-level processes executed by the NoC router.
$E_r(k)$	Total energy delivered to router r at cycle k.
E_ψ	Energy consumed when executing micro process ψ.
$\alpha_\psi(r,k)$	The number of occurrences of process ψ in router r at cycle k.
I_{ch}	The link's current profile for channel ch.
g	Power grid granularity multiple used to map the fine grid to a coarse grid.
l_f	The fine power grid segment length.
l_X	The coarse power grid segment length.
$Error_g$	V_{DD} drop computation error due to reduction in grid granularity.
V_X	Voltage for the coarse-grained grid model.
V_f	Voltage for the fine-grained grid model.
G_f	Fine-grained (original) power grid.
G_X	Coarse-grained (reduced) power grid.
$t_{clk\_Q}^i$	Clock-to-Q delay of latch i.
t_{setup}^i	Critical setup time of latch i.
$t_{wire}^{i,j}$	The delay in wire i,j.
$Pr(Err_i)$	Probability of error for link i.
γ_i	Utilization of link i.

circuitry, which means that link traversal power is supplied by the flit forwarding router and the repeaters along the link path. In our model, the workload of both routers and links are characterized by capacitance. This capacitance is determined by the charge delivered to the circuit during the switching time period.

Based on the above, the router capacitive load for router $r \in \{R\}$ (see Table 3.1) in the power grid at the k^{th} switching cycle ($C_r^{load}(k)$) is computed as:

$$C_r^{load}(k) = C_r^{links}(k) + C_r^{circuit}(k), \tag{3.3}$$

where $C_r^{links}(k)$ is load equivalent capacitance due to the link traversal of the flits at cycle k, and $C_r^{circuit}(k)$ is the load equivalent capacitance due to router circuit switching activity at cycle k. The capacitances here vary over time to reflect dynamic communication load changes in the network.

3.3.2.1 *Link Workload*

Link traversal load is the summation of loads for the links of all the channels in the router:

$$C_r^{links}(k) = \sum_{\forall ch \in \{CH_r\}} C_{ch}(k) \tag{3.4}$$

where, for a mesh topology, the set of router channels are $CH_r = \{North, East, South, West, Local\}$.

The load capacitance of the channel ch link at cycle k, $C_{ch}(k)$, can be computed from the link's current profile at that cycle as follows:

$$C_{ch}(k) = \frac{Q_{ch}(k)}{V_{DD}} = \frac{\int_0^{T_{clk}} I_{ch}(k,t)dt}{V_{DD}} \tag{3.5}$$

where $Q_{ch}(k)$ is the total charge delivered to the link of the channel at cycle k, $I_{ch}(k,t)$ is the current profile of the channel link at cycle k, and T_{clk} is the clock frequency period. In this work the link model proposed in [189] is employed to compute this current profile after modifying the formula for total channel link current I_{ch} to include wire switching.

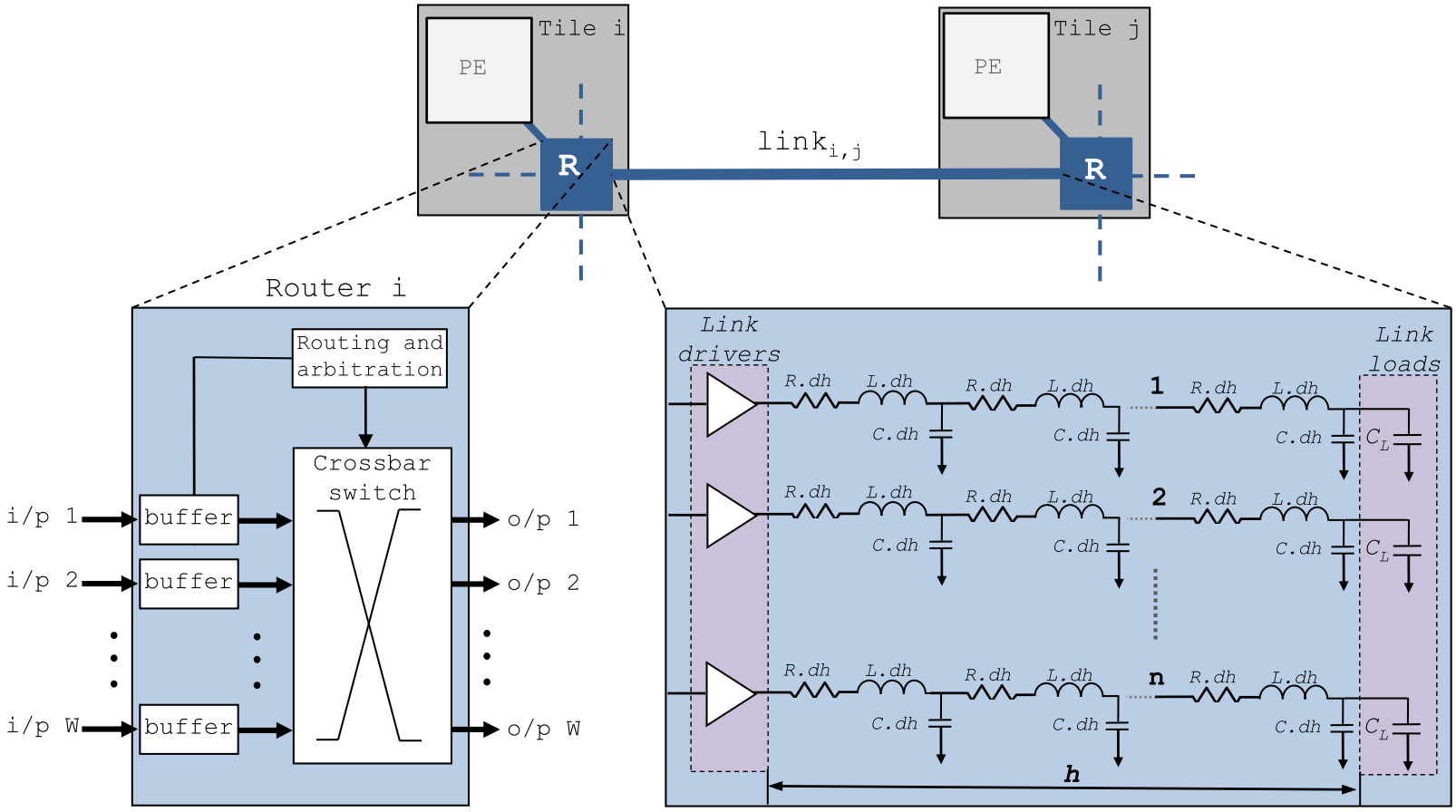


Figure 3.3: Illustration of the modelled NoC components including the routers connecting tiles i and j, with the router functional components, and NoC link, with its equivalent circuit assuming that router i is sending and router j is receiving. n: number of wires in the link, h: link length, W: number channels in the router.

Let *SW* be a vector of size n, which is equal to the number of link wires, with elements $sw_i \in \{0, 1, -1\}$ where 0, 1 and -1 represent quiet, switching up ($0 \rightarrow 1$), or switching down ($1 \rightarrow 0$), respectively. The time profile of the current draw of the channel link at cycle k, $I_{ch}(k,t)$, is given by:

$$I_{ch}(k,t) = \sum_{s=1}^{n} \sum_{i=1}^{n} M_{s,i}^{T} sw_{i,ch,k} \sum_{j=1}^{n} M_{i,j}^{T} I_{i,ch,k}(t) \tag{3.6}$$

where $I_{i,ch,k}$ is the current of wire $i \in \{1...n\}$, for the channel link at cycle k, M is the decoupling transformation matrix, and $sw_{i,ch,k}$ is the switching direction of wire i at cycle k [189].

3.3.2.2 *Router Workload*

The capacitive load of the router circuit for a switching time period can be computed from the energy it consumes during this time period. This energy is determined by integrating an NoC simulator, in order to determine the processes that are taking place in the router at each cycle, and a router power model which determines the energy consumed by each of these processes.

Let Ψ={RECEIVE, ROUTE, FORWARD, STANDBY} be the set of microarchitectural-level processes that can be executed by the router. These processes involve: receiving a flit, route computation, forwarding a flit, and no activity (static energy), respectively. The energy of the RECEIVE process is the energy required for writing to the input buffer. The ROUTE process energy is required for route computation, which is performed only for header flits for wormhole routing. The FORWARD process energy is the summation of the energies required for reading from the input buffer, switch allocation, switch traversal and link traversal. Also, let $\alpha(r,k)$ be the number of occurrences of process $\psi \in \Psi$ in router r at cycle k. Now, the total energy delivered to router r at cycle k, $E_r(k)$, can be expressed as:

$$E_r(k) = \sum_{\forall \psi \in \Psi} E_\psi . \alpha_\psi(r,k). \tag{3.7}$$

where E_ψ is the energy required by the router circuit to execute process ψ. This energy, for a particular router design, can be computed using a router microarchitectural power model while α_ψ can be determined using a cycle-accurate NoC simulator. The router load equivalent capacitance at cycle k ($C_r^{circuit}(k)$) can now be computed as follows:

$$C_r^{circuit}(k) = \frac{E_r(k)}{V_{DD}^2} \tag{3.8}$$

The load capacitance which results from Eq. 3.3 is used to characterize the load in Eq. 3.2.

The following sections are based on the assumption that router r is supplied with power through a set of nodes (G_r) in the power grid and, in line with previous works [85, 134], the resulting router load is divided equally over the set G_r. This set is determined by floorplan information, based on the geometrical structure of the grid and areas and positions of the routers.

3.3.3 *Power Grid Granularity*

Power grids designed for real VLSI circuits may contain tens of thousands or even millions of nodes [140], which results in impractical simulation time and memory requirements. Thus, a coarse grid approach has been used in many previous works [111, 118, 82], where a multi-grid based model order reduction is used and the number of nodes in the power grid is reduced by node elimination. The analysis is performed on the reduced coarse grid. Then, the solution is mapped back to the original (fine-grained) grid using linear interpolation, taking into account the values of conductances between the nodes.

During the mapping from the fine grid (G_f) to the coarse grid (G_X), the geometrical coordinates of the nodes in G_X (node j' in Fig. 3.4) must be the same as their equivalent nodes in G_f (node j in Fig. 3.4) in order to preserve the structure of the grid. For a regular mesh topology, this mapping takes the ratio ($g = \frac{l_X}{l_f}$) of the segment length of the fine grid, l_f, and the segment length of the coarse grid, l_X, as inputs (see Fig. 3.4). In order to for the total resistance to remain equal, a segment's width must be increased in proportion to its increasing length.

The relative error of the voltage drop for node $j' \in G_X$ due to grid granularity reduction can be expressed as:

$$\mathrm{Error}_g(j') = \left| \frac{\Delta V_f(j') - \Delta V_X(j)}{\Delta V_f(j)} \right| \times 100\%, \tag{3.9}$$

where ΔV_f and ΔV_X are the voltage drops in the fine and coarse grids, respectively, and j and j' are the nodes in the fine grid and its equivalent node in the reduced coarse grid, respectively.

In power grid simulation there are two techniques for model solution: iterative and direct [171]. Due to the very large grid size, the iterative technique is more suitable for analyses for which a single system solution is obtained, for instance the DC analysis of power grids. On the other hand, direct techniques are more convenient when several model solutions are necessary. This is the case for the present work, thus; a coarse-grained lumped model for the power grid is used in order to achieve the results in a practical simulation time.

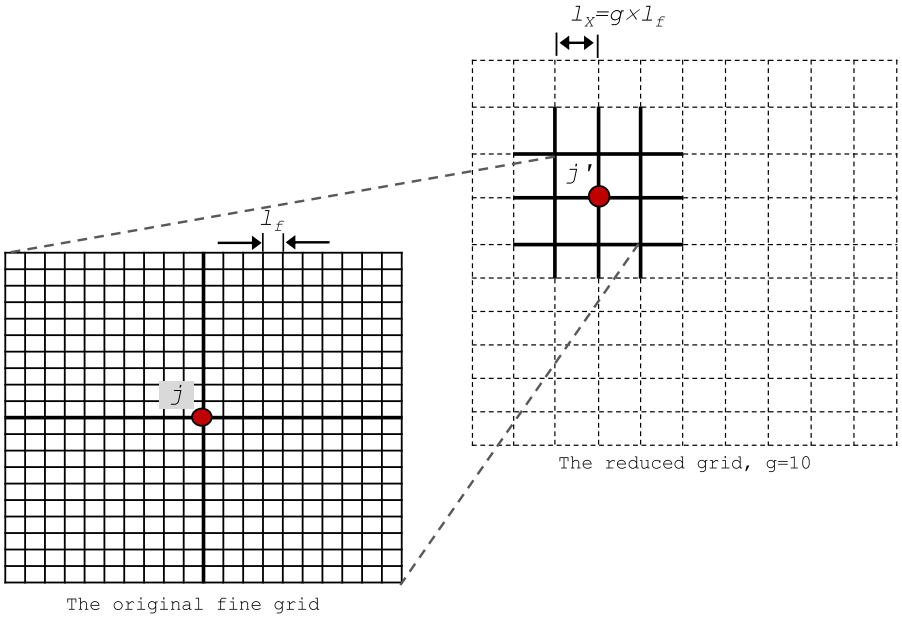


Figure 3.4: Illustration of the mapping from a fine-grained to a coarse-grained model of the power delivery grid.

3.4 EXPERIMENTAL RESULTS AND DISCUSSION

3.4.1 *Experimental Setup*

To evaluate our model, the floorplan and architecture of Intel's TeraFlop NoC [192] is adopted. A mesh NoC router is assumed with five channels and 39-bit communication links in each channel. The technology used is 65 nm with a frequency of 3 GHz and nominal V_{DD}=1V. Orion [112] is used for router power and area computation. Communication traffic simulation is conducted using Noxim [70], a SystemC-based NoC simulator. A Gaussian random distribution is assumed for the link switching activities, with a Poisson distribution for the packet injection. A uniform buffer size of 16 flits and a packet length of 3 flits are assumed. These values are in line with Intel's TeraFlop NoC configuration [192]. The models are integrated in an automated flow (Fig. 3.1) to compute the power supply voltage variations as a function of NoC switching activity. To evaluate our model, the floorplan and architecture of Intel's TeraFlop NoC [192] is adopted. A mesh NoC router is assumed with five channels and 39-bit communication links in each channel. The technology used is 65 nm with a frequency of 3 GHz and nominal V_{DD}=1V. Orion [112] is used for router power and area computation. Communication traffic simulation is conducted using Noxim [70], a SystemC-based NoC simulator. A Gaussian random distribution is assumed for the link switching activities, with a Poisson distribution for the packet injection. A uniform buffer size of 16 flits and a packet length of 3

flits are assumed. These values are in line with Intel's TeraFlop NoC configuration [192]. The models are integrated in an automated flow (Fig. 3.1) to compute the power supply voltage variations as a function of NoC switching activity.

For the power delivery network (PDN), a lumped model which includes both on-chip and off-chip power delivery network models is used. The on-chip PDN consists of a global-level mesh structure routed in the top metal layers. Unless otherwise stated, the on-chip power network is modelled as an RLC mesh with a grid segment length such that we have 5×5 granularity per NoC tile. Based on the analysis described in Section 3.4.3 below and as suggested by previous works [85, 115], this is sufficient for capturing the power supply voltage variations across the chip with reasonable accuracy and simulation time. The RLC values of the grid segments and link wires were determined using PTM [36].

3.4.2 *Model Verification*

Firstly, an experiment to evaluate the accuracy of the proposed model is performed. The power trace of a 3×3 NoC is computed under the $\mathtt{Transpose}$ traffic (in which $\mathtt{tile}(i,j)$ sends packets to $\mathtt{tile}(j,i)$), which is obtained from the NoC simulator for a packet injection rate of $0.015\ \mathtt{packets/cycle/node}$. Then, a SPICE netlist of the circuit is generated. In this netlist the workloads are modelled as triangular current sources. The peaks of these current sources are computed based on power traces taken from the router and link power models and the activity of the NoC. This enables the evaluation of the voltage variations resulting from integrating all the components of the model together (activity, power, and grid models). This circuit netlist is simulated in SPICE to obtain the resulting grid node voltages. The same scenario is also simulated using our model following the methodology described in Section 3.3. Power grid node voltages obtained from both the model and SPICE are presented in Fig. 3.5, which shows good matching with a mean relative error of only 4.7%.

3.4.3 *Granularity Analysis*

The impact of power grid granularity on both model accuracy and simulation time is also evaluated. A power grid for an area of $1\mathrm{mm}^2$ is simulated assuming a load of a 39-wire link and computed using Eq. 3.6. V_{DD} drop across the grid is determined with different power grid granularities. Coarser grids are generated by doubling the grid segment lengths and widths to preserve the resulting resistance.

Table 3.2 shows the results of this analysis for grid granularities starting from 40×40 (6400 nodes) down to 5×5 (25 nodes). Taking the

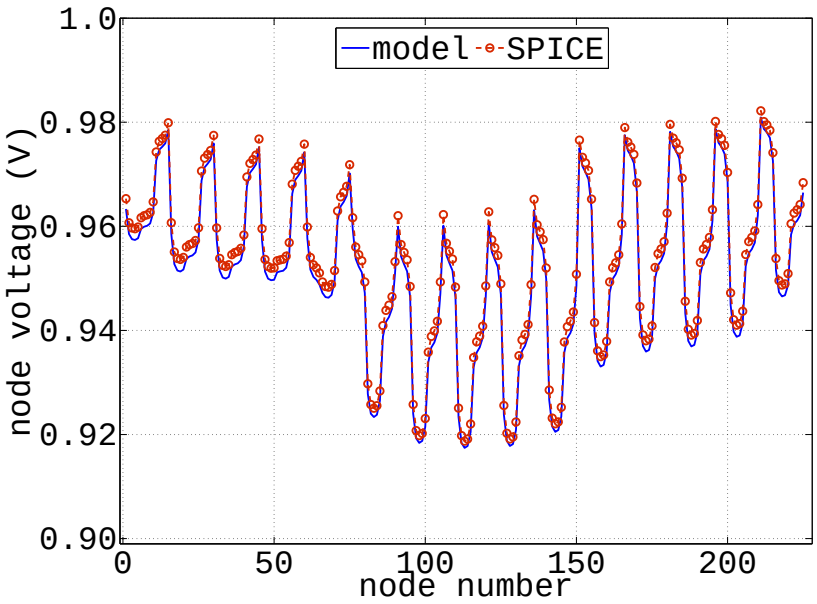


Figure 3.5: Comparison of node voltages computed by the proposed model with SPICE simulation for a 3×3 NoC configuration.

SPICE simulation of the 40×40 grid as a baseline, both relative error (Eq. 3.9) and simulation time are shown.

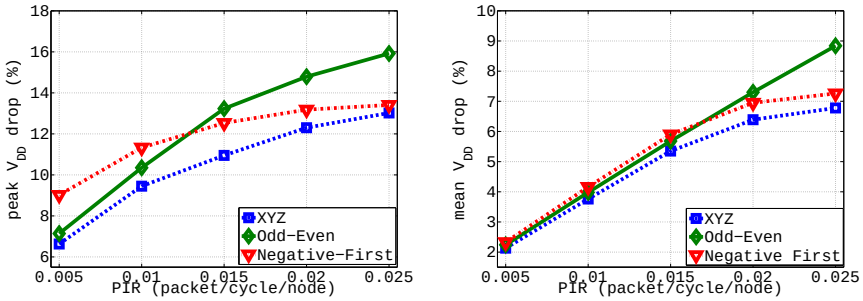
The results show that model accuracy decreases when the granularity of the model is quartered. On the other hand, simulation speed rapidly increases with granularity due to higher model order reduction. Considering fixed grid granularity per tile, the simulation time increases linearly with the NoC size (number of tiles) due to the fact that the number of power grid nodes increases linearly with the number of tiles.

l_X (μm)	l_f (μm)	#of nodes	$g = l_X/l_f$	Error_g (%)	time (s)	using
25	25	40×40 (1600)	1	-	8.9	SPICE
25	25	40×40 (1600)	1	1.98	1.152	model
50	25	20×20 (400)	2	6.07	0.155	model
100	25	10×10 (100)	4	8.6	0.022	model
200	25	5×5 (25)	8	11.85	0.006	model

Table 3.2: Comparison of the proposed model with different power grid granularities with the SPICE simulation of a fine granularity.

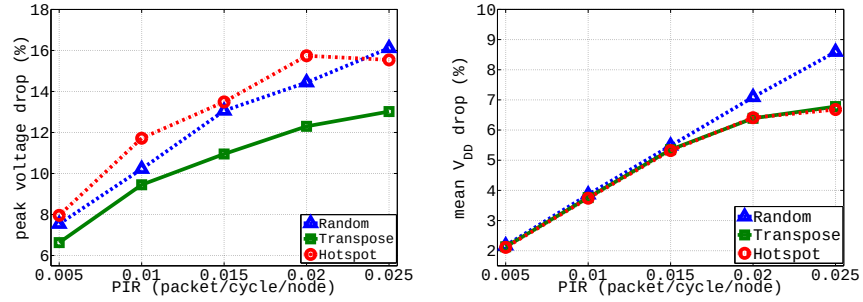
3.4.4 *Synthetic traffic patterns and Routing Algorithms*

In this section the results for PSN caused by different synthetic traffic patterns and routing algorithms are presented and discussed. A 6×6 NoC is considered here. traffic patterns used are *Random*, *Transpose*, and *Hotspot*. For Random traffic, each tile sends data to all other tiles with equal probability. For the Transpose case, $\mathtt{tile}(i, j)$ sends packets to $\mathtt{tile}(j, i)$. For the Hotspot traffic pattern the four central tiles receive an extra 5% in addition to the uniform (Random) traffic. Four routing algorithms; *XY*, *Odd-Even* (OE), *Fully-Adaptive* and *Negative-First* (NF) are also considered. The number of clock cycles necessary for captur-



(a) Peak V_{DD} drop (b) Mean V_{DD} drop

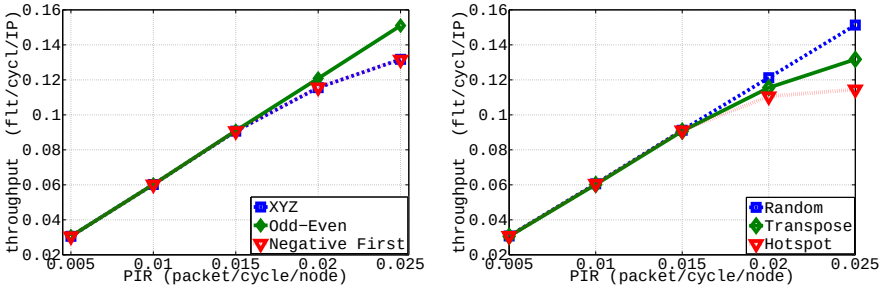
Figure 3.6: V_{DD} drop versus PIR for various routing algorithms.



(a) Peak V_{DD} drop (b) Mean V_{DD} drop

Figure 3.7: V_{DD} drop versus PIR for various traffic patterns .

ing the characteristics of the workload differs from one application to another. For synthetic traffic it should be noted that the power traces have been repeated after 10,000 clock cycles. However, the model is run for 100,000 clock cycles in order to guarantee the coverage of all workload characteristics. The resulting V_{DD} drop is plotted for different routing algorithms (Fig. 3.6) and traffic distributions (Fig. 3.7) for a range of packet injection rates. The peak and mean drops are shown in both figures. The throughputs achieved for these routing algorithms and traffic patterns are also shown in Fig. 3.8. Spatial V_{DD} drops un-



(a) (b)

Figure 3.8: Throughput for different (a) routing algorithms, and (b) traffic patterns.

der these traffic patterns and routing algorithms are given in Figures 3.9 and 3.10 respectively for a PIR of 0.015 packets/cycle/node.

Note that, in general, there is a considerable increase in V_{DD} drop with PIR. This is expected, since a higher packet injection rate leads to a higher throughput which increases the switching activity of the routers and data links. This, in turn, raises the current draw causing a higher V_{DD} drop. However, different routing algorithms and traffic patterns behave differently in terms of V_{DD} drop with increase in PIR. For instance, for low PIR (< 0.015) it can be noted that the NF routing algorithm causes higher peak (Fig. 3.6a) and mean (Fig. 3.6b) drops than the XY and OE, although it achieves the same throughput (and thus consumes the same power) within this PIR range (see Fig.3.8a). This is due to the fact that the NF algorithm tends to migrate the traffic to the negative quarter of the NoC mesh, as can be seen in Fig. 3.9d. This can create hotspots that would suffer higher supply drops due to unbalanced power density. On the other hand, both XY and OE have more balanced spatial workloads compared to NF, which leads to lower supply drops.

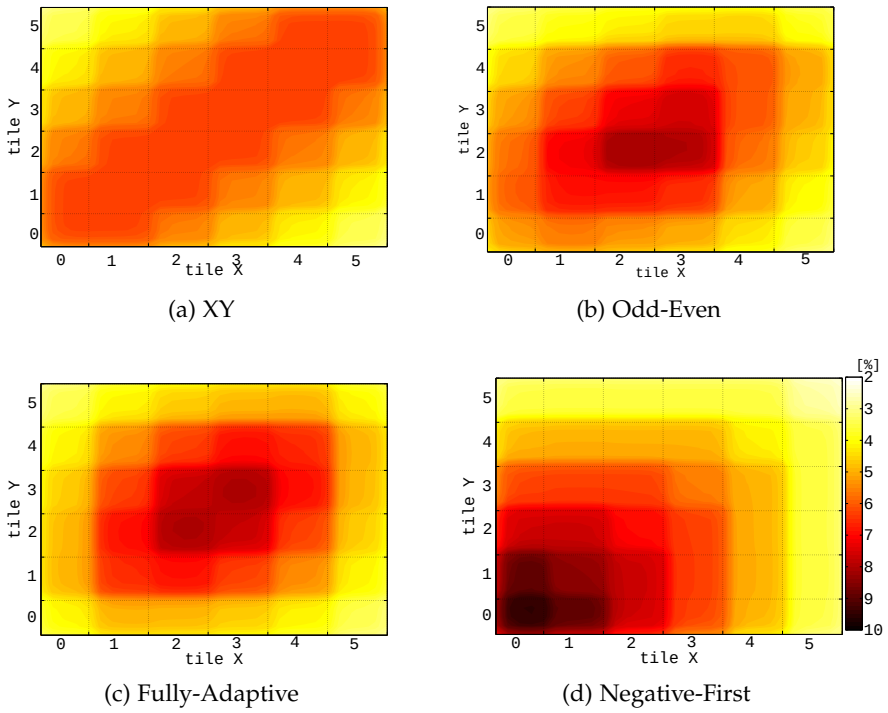
At high PIR (> 0.015) the V_{DD} drop for both NF and XY is less than that of OE due to the fact that the latter achieves higher throughput at this PIR range. This is because the NoC starts to saturate and throughput decreases for the XY and NF, which is not the case for OE (see Fig. 3.8a).

For traffic patterns, it can be seen that Hotspot traffic causes a higher V_{DD} peak drop, due to the centric nature of this traffic distribution and for the same reasons discussed above. This drop is reduced at higher PIR (> 0.015) due to the reduced throughput caused by saturation. Random traffic results in higher throughput in this range of PIR (Fig. 3.8b), which causes higher peaks and mean drops.

Figures 3.9 and 3.10 show the spatial distributions of selected routing algorithms and traffic patterns, respectively. In general, it can be observed that the power supply drop which results from a traffic pattern or routing algorithm is determined by the workload and increases

with this workload. Also, the spatial distribution of a traffic/routing workload plays an important role. Highly unbalanced traffic/routing can lead to significantly higher PSN compared to balanced cases, even when the workload is the same.

Table 3.3 summarizes the results for peak and mean V_{DD} drops for the set of traffic patterns and routing algorithms considered.



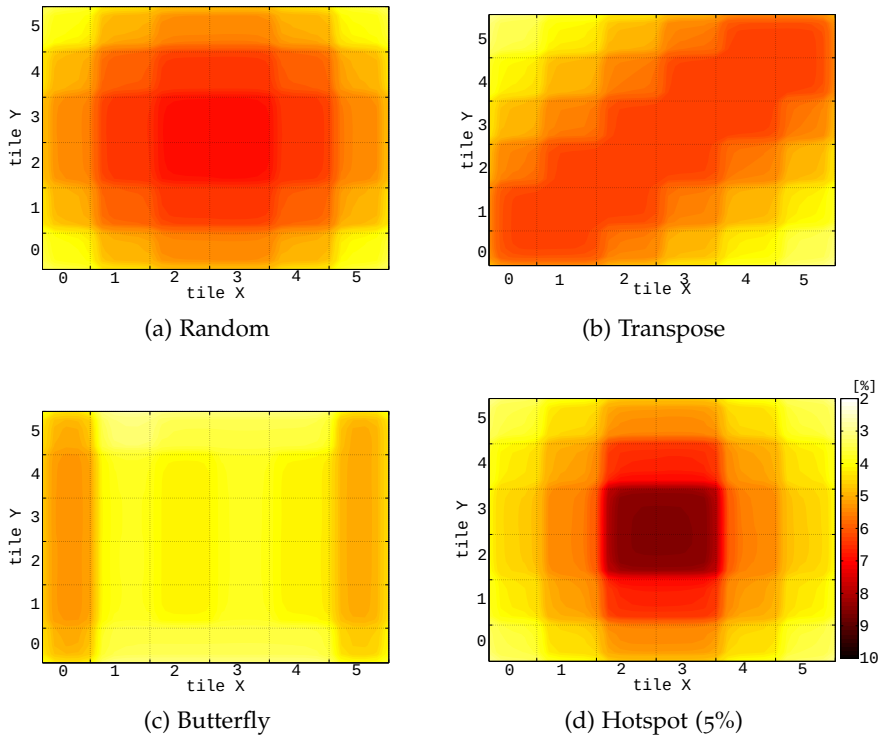
(a) XY (b) Odd-Even

(c) Fully-Adaptive (d) Negative-First

Figure 3.9: Spatial distribution of mean V_{DD} drop (%) for different routing algorithms and Transpose traffic.

	TRAFFIC					
	Random		Transpose		Hotspot	
ROUTING	peak	mean	peak	mean	peak	mean
XY	13.63	5.6	12.95	5.6	13.18	5.6
Odd-Even	11.51	5.48	13.82	5.5	12.53	5.45
Negative-First	14.15	5.31	13.81	5.31	14.21	5.3
Fully-Adaptive	12.96	2.92	13.56	5.48	12.79	5.32

Table 3.3: Summary of V_{DD} drop (%). Results of four different routing algorithms and three traffic patterns.



(a) Random (b) Transpose

(c) Butterfly (d) Hotspot (5%)

Figure 3.10: Spatial distribution of mean V_{DD} drop (%) for different synthetic traffic patterns with XY routing.

3.4.5 *Real Traffic*

To generate a realistic communication scenario, a generic complex multimedia system (MMS) is used. MMS comprised of an H263 video encoder, an H263 video decoder, an MP3 audio encoder, and an MP3 audio decoder [92]. Three mapping strategies were considered to map this benchmark to a 5×5 NoC: maximizing performance (minimizing packet latency) [137]; minimizing energy [92]; and random mapping. The resulting V_{DD} drops in the presence of the resulting three traffic patterns are computed using our tool. The power trace for this benchmark is found to be periodic, with a period of nearly 70,000 clock cycles. The simulations are run for 100,000 cycles to guarantee the coverage of workload characteristics.

Fig. 3.11 shows the spatial distribution of the power supply drop. It can be seen that performance and energy mappings are relatively close to each other in terms of V_{DD} drop. However, the energy-aware mapping has slightly higher peak drops compared to performance-aware mapping. It can also be seen that, in this instance of random mapping ,there is considerably higher drop compared to other types of mapping. This is caused not only by the higher power used for this mapping, but also the spatial distribution of this power profile which

results in higher power (and thus current) density in the central tiles, leading to a higher voltage drop.

3.5 CASE STUDY: POWER SUPPLY-AWARE TIMING ANALYSIS

The modelling of power supply noise in NoCs would have many applications in the design space exploration and evaluation of many-core systems at various design levels. Power grid integrity analysis, PSN-aware application mapping, and floor planning are examples of these applications. However, our tool can also be used to analyse the impact of the resulting V_{DD} variations on timing accuracy for circuit-dominated paths [163], or link-dominated paths, such as communication links and clock distribution networks [116, 45]. In this section, the developed model is employed in a technique for performing statistical timing analysis in the presence of power supply variations. This enables an estimation of timing violations in order to show how the resulting PSN affects timing and system reliability.

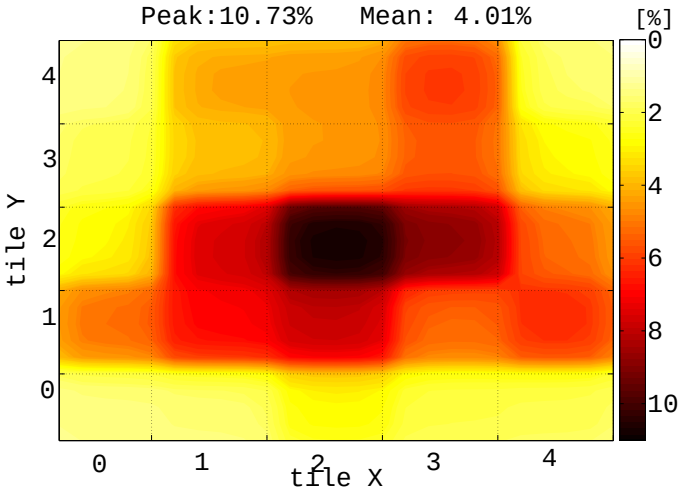
Power supply noise can cause significant increases in delay for circuit-dominated as well as interconnect-dominated paths [117, 116]. This delay could lead to violations of timing constraints in these paths and thus generate soft errors. It has been reported that, for these timing constraints to be met for a 20% supply variation, a 42% decrease in frequency is required for 65nm technology [116].

On the other hand, the International Technology Roadmap for Semiconductors (ITRS) predicts that there will be a significant gap between interconnection delay and gate delay. Moreover, this gap is expected to rise exponentially. This is particularly so for global interconnects where, even when repeaters are used, interconnect delay is expected to reach 9x the delay of the gates for a 32nm technology node, according to an ITRS [103]. This delay gap is expected to continue growing, leading to severe challenges in designing interconnects that comply with the timing constraints.

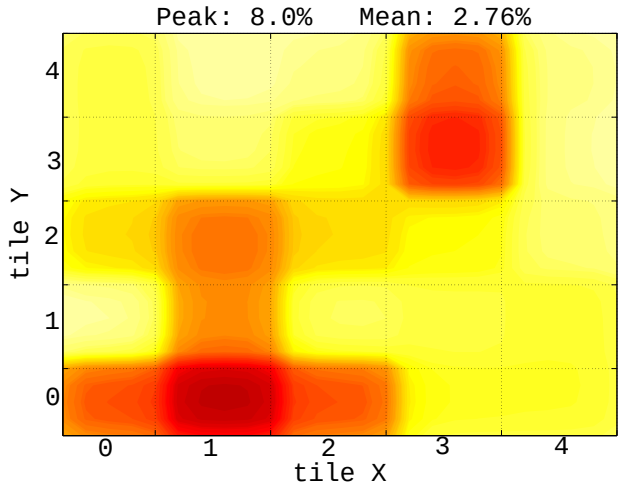
These factors motivate performing statistical timing analysis for NoC interconnects in the presence of power supply variations. This enables an evaluation of the probability of timing violations and BER due to PSN.

3.5.1 *Impact of Power Supply Variations on Link Delay*

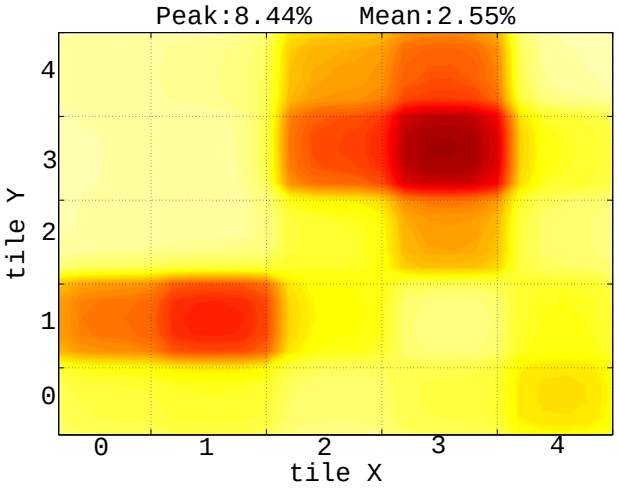
A major impact of PSN on performance can be seen by its impact on delay. Power supply voltage drops can cause significant increases in the delay of circuits and interconnects [117, 116]. This delay could lead to violations of timing constrains in these paths leading to soft errors. To compute the probability of switching error due to timing delays under power supply variations, a full knowledge of the V_{DD} variation distribution is needed. Also, the relationship between delay



(a) Random mapping



(b) Maximum performance mapping



(c) Minimum energy mapping

Figure 3.11: Spatial distribution of mean V_{DD} drop (%) for the MMS application traffic with three mapping strategies.

and V_{DD} for various components of the link is necessary. The delay

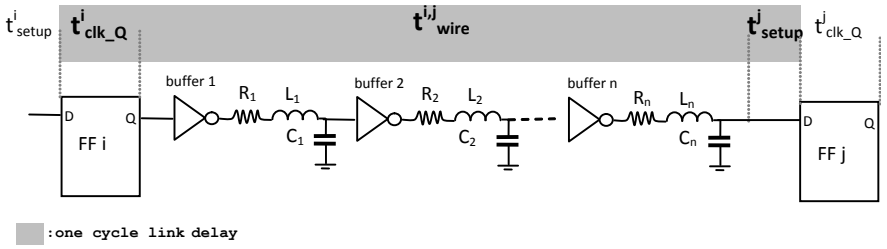


Figure 3.12: A model of on-chip link illustrating the delay components and timing constrains [158].

components of on-chip global interconnects are illustrated in Fig. 3.12. Considering a synchronous data path between two tiles, i and j, of a NoC data link and assuming a zero clock skew between these two, the sum of the clock-to-Q delay of the sending flip-flop (FF), $t^i_{clk\_Q}$, plus the wire delay between the two FFs, $t^{i,j}_{wire}$, in addition to the setup time of the receiving FF (t^j_{setup}) must not exceed the link clock period T_{clk} [123, 158]:

$$t^i_{clk\_Q} + t^{i,j}_{wire} + t^j_{setup} < T_{clk}. \tag{3.10}$$

Violation of this timing constraint would lead to switching errors in the link.

To perform timing analysis, and in line with previous works [163, 117], a quadratic approximation is adopted to determine the impact of V_{DD} drop on these delay components:

$$t_d(\Delta V_{DD}) = k_1 + k_2(\Delta V_{DD}) + k_3(\Delta V_{DD})^2 \tag{3.11}$$

where, $t_d(\Delta V_{DD})$ represents any of the link timing components on the left hand side of Eq. 3.10 and k_i (i=1,2,3) are technology-dependent constants. Assuming 65 nm technology, an edge triggered D-FF, which comprises two master-slave D latches, is simulated in SPICE and the clock-to-Q and setup times of the FF under V_{DD} variations are obtained. The wire delay is also obtained for a 2mm length with repeaters chosen according to the results in previous work [123]. Fig. 3.13 plots the delay obtained for t_{clk_Q}, t_{wire} and t_{setup} when the V_{DD} drop is varied from 0% ($V_{DD} = 1.0V$) to 25% ($V_{DD} = 0.75V$) of nominal supply voltage with steps of 5% (50mV). Using these results, analytical formulas relating the clock-to-Q delay, setup time and wire delays to the V_{DD} drop are obtained using regression.

3.5.2 *Probability of Timing Violation Errors and Bit Error Rates*

Using our model, the distribution of V_{DD} are obtained for all chip components. In general, the V_{DD} distribution obtained consists of

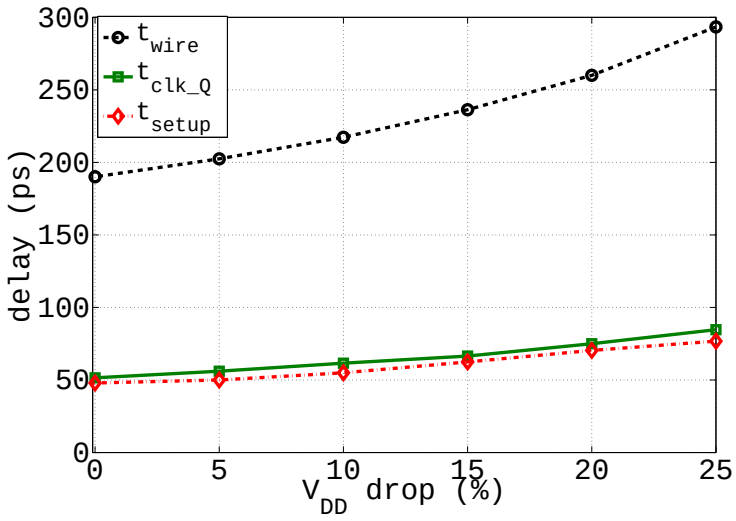


Figure 3.13: The t_{wire}, t_{clk_Q} and t_{setup} link delays versus V_{DD} drop.

a DC component (or IR) drop, which results in the shifting of the mean of the V_{DD} distribution, and an AC component (or ΔI drop) which results in variation in V_{DD}. Using the formulas for link delay (Eq. 3.11), a statistical timing analysis can be performed to obtain the distribution of the resulting variations in link delay due to power supply variations for all NoC links. Thus, the IR drop translates into a delay skew and the ΔI drop translates into delay jitter for these links.

For FFs, (t_{clk_Q}) and t_{setup} are computed from the V_{DD} of the sending and receiving tiles respectively. The wire delay (t_{wire}) is computed using the V_{DD} between the sending and receiving tiles.

The resulting delay distribution of a link can be used to estimate the probability of timing error due to power supply variations for that link. This probability is estimated as the portion of the delay distribution that does not satisfy the constraint in Eq. 3.10. In other words, the probability of timing error on link l, $Pr(Err_l)$, can be expressed as:

$$Pr(Err_l) = Pr(t_l > T_{clk}) \tag{3.12}$$

where t_l is the total link delay which is computed in the presence of V_{DD} variations using Eq. 3.11.

Fig. 3.14 shows the results of this analysis for the MMS benchmark with maximum performance mapping. Fig. 3.14a shows the distribution of delay means (skews) for all links and Fig. 3.14b shows the distribution of delay standard deviations (STDs) or jitters for these links. It is found that links with the highest probability of error belong to the tile which suffers the highest V_{DD} drop. This is tile 2, as can be seen in Fig. 3.11b.

Using the probability of errors for each link, the average BER for the NoC can be computed. Given the application and mapping characteris-

tics, let the relative utilization of NoC link l be γ_l, which is the ratio of the data volume communicated through this link (v_l) to the total data volume communicated in the NoC. Thus, the relative utilization of the link can be characterized as:

$$\gamma_l = \frac{v_l}{\sum_{\forall i,j \in \{N\}} v_{i,j}} \tag{3.13}$$

where $\{N\}$ is the set of all nodes (tiles) in the NoC and $v_{i,j}$ is the data volume that needs to be communicated between tile i (as a source) and tile j (as a sink). Now, the BER for all the NoC links (assuming that the links are independent) can be computed as follows:

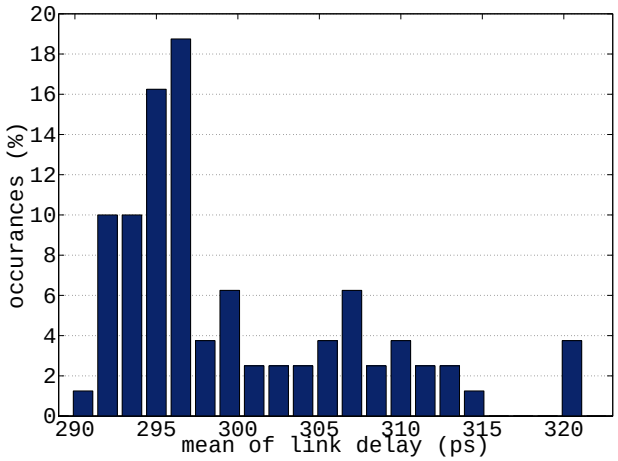
$$BER = \sum_{\forall l \in \{L\}} \gamma_l . \alpha_l . Pr(Err_l) \tag{3.14}$$

where $\{L\}$ is the set of all links in the NoC and α_l is the average switching activity of link l. The switching activity α_l is the average spatial activity of the link, which is determined by the average hamming distance between consecutive flits; while γ_l can be seen as the average switching activity over time. Both are in the range of 0-1.

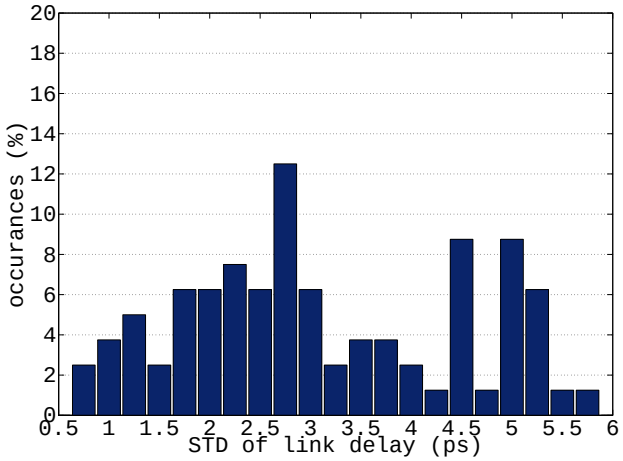
For the MMS benchmark with performance mapping, and assuming α_l is 25% the average BER is found to be 5.95×10^{-6}. On the other hand, when random mapping is considered, the NoC experiences a higher drop (see Fig. 3.11) and BER increases to 2.9×10^{-5}.

To illustrate the impact of increased communication workload in terms of network throughput on BER for various traffic scenarios, Fig. 3.15 plots BER for different synthetic traffic patterns against throughput. It can be noted that BER increases exponentially with throughput for all traffic patterns. However, due to different hotspot and traffic distributions, different traffic patterns experience different BERs. The higher and more concentrated the traffic patterns, the higher the BER, as can be seen for the Hotspot traffic in Fig. 3.15.

It worth mentioning here that the above timing analysis is pessimistic and, in practice, no hardware system would tolerate such error rates. The reason for the above results is that our analysis does not consider many design precautions that are normally taken to prevent timing violations. These precautions would usually lead to a "guard-gap" which ensures that timing violations rarely happen. Thus, this timing analysis and the resulting BER is to be used here as comparative metric only to compare various application mapping strategies . In particular, it is used to compare the timing accuracy of activity-balance mapping with energy-aware mapping and is not, by any means, an accurate timing analysis of the resulting systems.



(a) Distribution of links delay (mean)



(b) Distribution of links delay (std)

Figure 3.14: Links delay statistics for the MMS benchmark using maximum performance mapping.

3.6 SUMMARY AND CONCLUSION

In this chapter, an integrated modelling tool to capture the impact of on-chip communication workloads on power delivery grid is presented. This tool is dedicated for NoCs. It integrates a NoC simulator, on-chip link model, NoC power and area models, and a fast power grid model to provide a comprehensive simulation and system analysis. The granularity of the power grid model affects the degree of accuracy of the analysis. Compared to the SPICE simulation results, the error of the power grid model is less than 2% and increases linearly with the granularity of the grid. The developed tool also provides a detailed analysis of power supply variation based on traffic distribution and routing algorithms. The practicality of the proposed model is

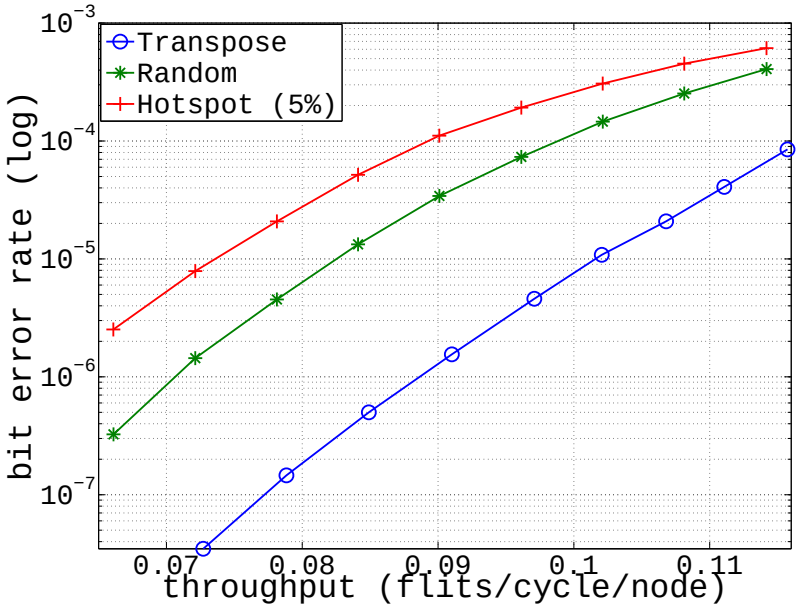


Figure 3.15: Bit error rate versus throughput for various synthetic traffic patterns.

further exemplified through a case study. Detailed statistical timing analysis of communication link delay is presented. This enables the study of the impact of PSN based on communication delay for different communication workloads and traffic patterns. The results from such analyses can be used to determine high-level performance metrics, such as the probability of error and bit error rates. Comprehensive analyses of the power grid and accurate communication link models are crucial to the evaluation of power supply integrity. The models and techniques proposed in this work enable a robust evaluation of NoC-based multi-core systems in the early design stages.

4

POWER SUPPLY NOISE MINIMIZATION THROUGH MAPPING IN NOCS

4.1 INTRODUCTION

Power integrity is crucial for reliable VLSI systems, and PSN has adverse effects on digital circuit performance and reliability. PSN could cause signal deterioration and create soft errors. Variations in power supply are found to have significant impacts on operational frequency and system power dissipation [141]. It has been reported that a fluctuation of 10% in power supply causes about 6% of change in frequency and 18% increase in power dissipation due to increased leakage [163]. However, it's becoming increasingly challenging to guarantee power integrity with the increasing scale of integration. This is due to higher power density and operating frequencies that result in continuously increasing PSN in the chip.

On the other hand, NoCs have been proposed as a promising communication paradigm for SoCs and CMPs. The NoC tackles many limitations associated with bus on-chip connectivity by providing a scalable, flexible and power-efficient solution to integrate many cores in a single chip [23]. However, networks-on-chip can produce considerable PSN around regions with high communication, particularly, with the communication centric architectures and/or applications in which the NoC consumes a substantial portion of system power. Moreover, the advances in both technology and architecture of future many-core systems would cause NoC power consumption to increase [33].

Due to the above-motioned factors, the characterization and optimization of communication workload, in terms of NoC load and power dissipation, is becoming increasingly necessary in order to ensure power integrity. Power dissipation activity in NoCs is strongly correlated with network traffic load. This load is determined by the communication bandwidth among application tasks and the way these tasks are mapped to the target NoC architecture. Many of the application mapping strategies for NoCs proposed so far focus on minimizing energy, which results in high activity tasks being placed close to each other. This results in high activity hotspots that experience significant voltage drops and temperature rises. In this chapter, an NoC application mapping strategy is proposed which aims to provide activity balancing and the minimization of PSN, . The main contributions of this chapter can be summarized as follows:

1. A new concept of optimizing PSN in application mapping for NoCs is presented which considers the impact of communication workload on the power delivery network for NoC-based systems.
2. The metric of activity density is introduced and the impact of its spatial patterns on power delivery is analysed. Significant relationships between the spatial distribution of core activity and PSN are discovered.
3. The tile repulsive force is proposed as a mapping strategy objective which, in contrast to other NoC mapping strategies, results highly active tiles being spread across the chip. This and causes a significant drop in PSN with low energy and performance overheads. Moreover, high reduction in PSN is achieved even when the system technology is scaled down.
4. Statistical timing analysis is performed for the resulting systems, showing that better timing accuracy is achieved by the new mapping strategy due to reduced hotspots and PSN.

The rest of this chapter is organized as follows: Section 4.2 describes related work on PSN mitigation and NoC application mapping. This section also explains the notations, definitions and the main models introduced in this chapter. In Section 4.3 the methodology used to achieve the minimization of PSN by application mapping in NoCs is presented. Experimental results and analyses are presented and discussed in Section 4.4. Finally, the conclusions of the chapter are given in Section 4.5.

4.2 RELATED WORK AND BACKGROUND

4.2.1 *Related Work and Motivation*

Classical techniques employ on-chip decoupling capacitors to reduce PSN. Additional capacitors are inserted in parallel with the power delivery networks to filter current spikes and reduce power supply fluctuations. This can be an effective technique at the cost of additional chip area. Various methodologies have been proposed to optimally determine the physical location for the decoupling capacitors during floor-planning [39].

For systems with power gating, gated blocks can produce significant PSN when switched on [85, 197]. In [85], a distributed power-delivery model has been proposed and used to analyse on-chip power supply variations in order to understand the impact of inter-core interactions in a CMP with power gating. It has been found that powering on all cores simultaneously can lead to significant voltage drops in the system. The authors proposed to resonate cores out-of-phase in order to reduce voltage swings. In another work [197], the authors proposed

an optimal power-gating techniques coupled with dynamic scheduling to minimize the voltage drop caused by high-frequency logic switching in the gated blocks.

PSN can also be mitigated at the application-level. In particular, application workload assignment can have a significant impact on the PSN induced in a multi-core system. In [185], a simulated-annealing approach has been employed to optimize the assignment of workloads to the cores, such that the resulting PSN can be minimized.

Although these works are shown to be efficient in reducing the overall PSN, the cores are assumed to run tasks independently. Thus, task-dependence and communication between the cores has been ignored.

On the other hand, there have been several attempts to model the impact of V_{DD} variations on the timing of circuits and links in VLSI systems [22, 116, 163]. The impact of PSN on performance in microprocessors for both link- and circuit-dominated paths has been analysed [163]. The authors showed that PSN would cause a frequency penalty of 6.7% for 130 nm, and that this penalty would significantly increase in future technology generations. In other works [22] and [116], models have been proposed to capture the impact of PSN on propagation delay through statistical timing analysis. These analyses would also enable the detection of any violations of timing constraints and critical timing paths.

Networks-on-chip can produce considerable PSN around the regions with high communication activity. This is particularly true when the NoC takes a substantial portion of system power; for example 40% in the MIT RAW chip [181] and 30% in the Intel 80-core TeraFLOPS chip [192]. However, the impact of PSN and the associated risks to an NoC system has been so far largely ignored. This is exacerbated by aggressive energy [137] and performance [94] focused optimization. This is because core with high communication are usually mapped to regions in a close proximity, and this is likely to lead to some regions in a chip having significantly higher activity density. More power would be drawn into these areas at the expense of voltage drops and PSN. This under-balanced communication patterns can be seen as an advantage in terms of energy and performance, but at the cost of system reliability.

The concept of activity density in VLSI circuits was first introduced by Najm [139]. The author define transition density to be the *average switching rate* and he developed an algorithm to compute this based on stochastic models of logic signals. The results showed that higher activity density will lead to higher power and ground current densities, which would directly affect power supply integrity and introduce thermal hotspots. Also they show that higher activity would have a negative impact on circuit reliability in terms of electro-migration

failures. This can be seen as another problem associated with higher activity density, which motivates activity balancing in VLSI systems.

In this chapter, a new NoC mapping strategy which aims at balancing switching activity and power density is presented. This mapping considers communication workload and interactions among the mapped application tasks. The new mapping improves power supply integrity through considerable reductions in PSN. This is attributed to the removal of hotspots and minimization of switching activity density.

4.2.2 *Background and Definitions*

This section, gives definitions and explains the notation used in this chapter, and briefly describes the metrics of PSN and energy. A summary of all notation used in this chapter is also shown in Table 4.1. Although this work is not limited to any particular NoC topology or architecture, for convenience it is described in the context of a regular mesh NoC.

4.2.2.1 *Definitions*

The NoC architecture is characterized by the *Architectural Graph* $\mathrm{ARG}(T, P)$ which is defined as a directed graph where each vertex, $t_i \in \{T\}$, represents an NoC tile and each directed arc $p_{i,j} \in \{P\}$, represents the path from tile i to tile j. Each path $p_{i,j}$ consists of a set of n links $L(p_{i,j})$, with the first link l_0 starting with tile i and the last link l_{n-1} ending with tile j. The set $L(p_{i,j})$ is determined by the routing algorithm used to route packets from source to destination.

On the other hand, the application's communication requirements are described by the *Application Graph* $\mathrm{APG}(S, A)$, which is a directed acyclic graph where each vertex $s_i \in S$ represents a task and each arc $a_{i,j} \in A$ represents the communication from task s_i to task s_j. The quantities associated with $a_{i,j}$ are the bandwidth requirement from s_i to s_j, $b(a_{i,j})$ and data volume $w(a_{i,j})$.

A mapping function (Ω) maps an application characterized by the APG to the target architecture characterized by the ARG.

4.2.2.2 *Metrics for Power Supply Noise*

Power supply noise typically refers to any fluctuations from the nominal supply voltage. However, in this thesis we need to quantify this noise and have a computable metric to account for it. In practice, not any voltage fluctuations is accounted for in noise computation. Typically, only voltage drops below a particular level, called noise margin, are considered. Also, the duration of the drop, and not only its value, has impact on system reliability and need also to be considered. Thus, in [48], the power supply noise at a power grid node $x \in G$, $\mathrm{PSN}(x)$,

Table 4.1: Definitions and notation used in this chapter.

Symbol	Definition
$ARG(T, P)$	NoC architectural graph with the set of NoC tiles {T} as vertices and the set of paths {P} among these tiles as arcs.
$APG(S, A)$	Application graph with the set of tasks {S} as verticies and the set of communications {A} among these tasks as arcs.
$L(p)$	Set of NoC links that constitutes a path p in the architectural graph.
Ω	Mapping function that maps the application graph to the architectural graph.
PSN_{tot}	Total power supply noise.
V_{NM}	Power supply noise margin.
$E_{tot}(\Omega)$	Total energy cost of mapping Ω.
W_r	Channel width of router r.
N_r	Number of channels in router r.
F_r	Frequency of router r.
$B_r(\Omega)$	Bandwidth capacity of router r.
$\hat{B}_r(\Omega)$	Bandwidth load of router r.
α_u	Activity factor of VLSI unit u.
$\gamma_k(D)$	Regional activity density of tile k for a region of diameter D.
Γ_i	Metaphor for charge of tile i, which is a function of the tile local activity density and given as follows: $\Gamma_i = e^{k_i \gamma_i}$, where k_i is a tuning constant.
$F_{i,j}$	Tile repulsive force between tiles i and j.
F_{tot}	Total repulsive force for all NoC tiles.
GD	Application graph edge density.
$\overline{BW}$	Application graph average bandwidth.
$t^i_{clk\_Q}$	Clock-to-Q delay of latch i.
t^i_{setup}	Critical setup time of latch i.
$t^{i,j}_{wire}$	Delay of wire i, j.
$P_{err}(l)$	Probability of error for link l.

is computed as the time integral of the drop below the noise margin voltage (V_{NM}) as follows:

$$PSN(x) = \int_0^{T_s} \min\left[V_{NM} - \Delta V(x,t), 0\right] dt \tag{4.1}$$

where V_{NM} is the noise margin, and $\Delta V(x,t)$ is the power supply variation for node x at time t. Total PSN on the chip is determined as the summation of the PSN for all nodes in the power grid:

$$PSN_{tot} = \sum_{\forall x \in G} PSN(x) \tag{4.2}$$

or

$$PSN_{tot} = \sum_{\forall x \in G} \left[\int_0^{T_s} \min\left[V_{NM} - \Delta V(x,t), 0\right] dt\right] \tag{4.3}$$

where PSN_{tot} is the total PSN on the chip. It can be noted that PSN is a time integral of voltage drop and thus its unit is volt.second ($v.s$).

4.2.2.3 *Energy Model*

To compute the energy cost of a mapping function, we used the bit energy metric is used. The average energy consumed by sending one bit from tile i to tile j ($E^{bit}_{i,j}$) is given by [206]:

$$E^{bit}_{i,j} = (E^{bit}_S + E^{bit}_B) n_{hops}(i,j) + E^{bit}_L (n_{hops}(i,j) - 1) \tag{4.4}$$

where $n_{hops}(i,j)$ is the number of hops along the path between tiles i and j. E^{bit}_S, E^{bit}_B and E^{bit}_L are the per-bit energy dissipated by the crossbar switch, buffer and link of the router, respectively. The total energy cost of a mapping function ($E_{tot}(\Omega)$) is computed as:

$$E_{tot}(\Omega) = \sum_{a_{i,j} \in \{A\}} w(a_{i,j}).E^{bit}_{\Omega(i),\Omega(j)} \tag{4.5}$$

where $w(a_{i,j})$ is the communication data volume from task s_i task to s_j in APG. The number of hops from the source to destination tiles i and j, $n_{hops}(i,j)$ is determined by the routing algorithm. In this work the, XY routing algorithm is assumed. XY is a deterministic and deadlock-free routing algorithm which routes the packets from their sources along the X direction first and then along the Y direction towards their destinations [66]. $\Omega(i)$ is the mapping of task i; that is, the tile that task i is mapped to by the mapping function Ω.

4.3 METHODOLOGY

4.3.1 *Local and Regional Activity Densities*

In this work, a mapping strategy based on minimizing activity density is employed. The metric of *local activity density* is first defined, considering an NoC tile k which consists of a set of functional units, such as router, floating point unit and SRAM. For unit $u \in k$, the local activity density (γ_u) is defined in terms of the unit's maximum power consumption (P_u^{max}), switching activity factor (α_u) and area as follows:

$$\gamma_u = P_u^{max} \left(\frac{\alpha_u}{area_u} \right) \tag{4.6}$$

where α_u ranges from 0 to 1 and is the ratio of the unit's load to its maximum loading capacity.

Now, considering an NoC tile k which consists of a set of the units, tile's local activity density, γ_k, can be expressed as:

$$\gamma_k = \frac{\sum_{\forall u \in k} P_u^{max} \alpha_u}{area_k} \tag{4.7}$$

where $area_k$ is the tile area.

The *regional activity density* of an NoC tile can be defined in terms of local activity densities. For a regular mesh NoC, a region with a particular size in the vicinity of a tile of interest k can be defined as the set of all tiles j that satisfy $d_{k,j} \leqslant D$. Here $d_{k,j}$ is the Manhattan distance between tiles k and j and D is the radius of the region. Now, considering a regular NoC with homogeneous tile sizes and architectures, the regional activity density of tile k as a function of region radius D, $\gamma_k(D)$, can be expressed as the summation of local activity densities of all tiles in this region:

$$\gamma_k(D) = \sum_{\forall j \in \{T\} \mid d_{k,j} \leqslant D} \gamma_j \tag{4.8}$$

According to this definition, the local activity density defined in Eq 4.7 can also be denoted as $\gamma_k(0)$. For topologies other than the mesh, other distance measures, such as physical distance, can be used to define the region.

Considering an NoC router, the switching activity is fully characterized by the communication demand of the application and the placement of the communicating tasks across the NoC system. For a router r with N_r number of channels, W_r channel width and F_r frequency, the bandwidth capacity r, B_r, can be expressed as:

$$B_r = W_r \times N_r \times F_r \tag{4.9}$$

The actual switching load of the router's logic, $\hat{B}_r$, is the bandwidth of data the router is responsible for relaying. This load is determined by the application mapping function (Ω) and the routing algorithm. The router load is the summation of the loads of all router links. For router r with a set of channels ($\{CH_r\}$), the router load is given by:

$$\hat{B}_r = \sum_{ch \in \{CH_r\}} \hat{B}(l_{ch}) \tag{4.10}$$

where $\hat{B}(l_{ch})$ is the load communication bandwidth of channel ch link, l_{ch}. The activity factor of router r, α_r, can be readily computed as the ratio of the router communication load to its bandwidth as follows:

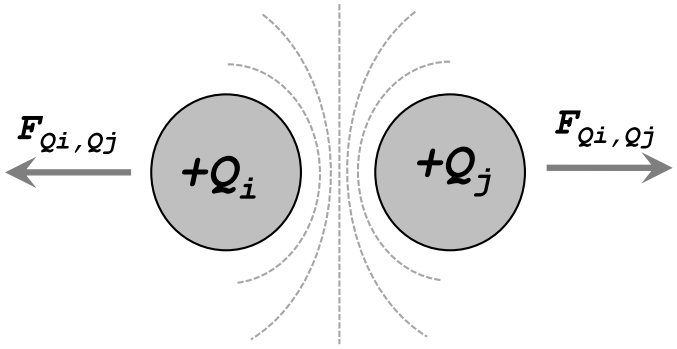
$$\alpha_r = \frac{\hat{B}_r}{B_r} \tag{4.11}$$

For tile units other than the router, such as the FPU, instruction memory and data memory, various techniques can be used to estimate the activity factor. Examples include using probabilistic methods and methods that consider the spatial and temporal correlations of the circuits in the switching probability model [25, 76]. However, the present work focuses on communication-centric applications in which computation is modulated by communication. Thus, the activity factor of tile units other than the router is estimated using the communication demand of the task assigned to the tile.

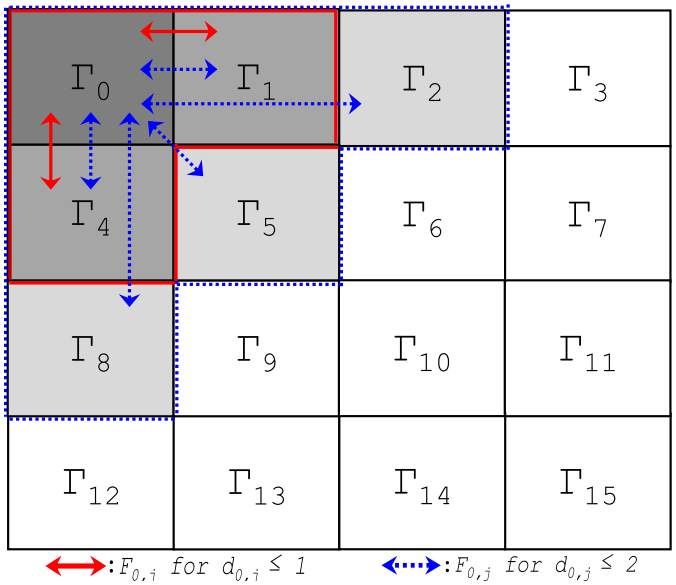
4.3.2 *Power Supply Noise Optimization*

Higher switching activity densities lead to higher average power and current densities. Moreover, high activity densities would also increase peak current demand due to larger amount of circuitry switching simultaneously. This causes both the average and peak resistive (IR) drops to increase. The inductive drop, ΔI, also increases with activity density due to higher rates of switching which lead to higher fluctuations in the current drawn [139, 17, 143]. Thus, minimizing activity density improves supply integrity and reduces PSN. These facts are also supported by the analysis of the correlations between PSN and both local and regional activity densities presented in Section 4.4.2 bellow.

To reduce the regional activity density within a particular region, the objective function must minimize the number of high activity tiles in this region (see Eq. 4.8). Thus, mappings that result in condensing high activity tiles must be penalized. To define this cost, the metric of *tile repulsive force* ($F_{i,j}$) is introduced. It is assumed that two tiles, i and j, repel each other by a force which is directly proportional to their



(a) Repulsive force among like charges



(b) Repulsive force between tile o and other tiles for different region sizes. $\Gamma_i = e^{k_i \gamma_i}$ is the tile charge.

Figure 4.1: Illustration of tile repulsive force.

local activity and inversely proportional to the square of the distance between them. This force can be expressed as follows:

$$F_{i,j} = K \frac{\Gamma_i \Gamma_j}{d_{i,j}^2} \tag{4.12}$$

where $\Gamma_i = e^{k_i \gamma_i}$, $\Gamma_j = e^{k_j \gamma_j}$ and k_i, k_j and K are constants. The exponential function is used to impose an exponential increase in $F_{i,j}$, and thus a higher cost, when local activity increases. The distance $d_{i,j}$ considered here is the Manhattan distance between the tiles.

This force is defined in Eq. 4.12 reflects strong dependency between spatial proximity of tile activities. The square in the denominator is used introduce a quadratic increase in this force with distance and is inspired by the physical phenomenon described by Columb's low. Moreover, the notion of force-directed optimization in microelectronic systems design is introduced in previous works where a force-directed

floorplanning techniques were used to optimize parameters such as area, timing, congestion and heat distribution during floorplanning [68, 47].

Since the force defined by Eq. 4.12 decreases quadratically with distance, and to reduce the computation time required to evaluate this force during mapping optimization, the repulsive force is considered among tiles lying of a region with a predefined size where the distance between tiles does not exceed a defined radius D, as illustrated in Fig. 4.1b. Thus, the total of this repulsive force is computed as:

$$F_{tot} = \sum_{\forall j \in \{T\}} \left(\sum_{\forall i \in \{T\}, d_{i,j} \leqslant D} F_{i,j} \right) \tag{4.13}$$

A mapping function which results in a lower value of F_{tot} will have a more scattered distribution of high activity tiles, reducing activity density and PSN.

4.3.3 *Problem Formulation*

The problem of PSN optimization mapping in NoCs can now be formulated as follows:

Given: Application graph, $APG(S, A)$ and architectural graph, $ARG(T, P)$ that satisfy:

$$|S| \leqslant |T| \tag{4.14}$$

find: A mapping function Ω that maps each task in APG to a tile in ARG

Objective:

$$\min \left[\sum_{\forall j \in \{T\}} \left(\sum_{\forall i \in \{T\}, d_{i,j} \leqslant D} F_{i,j} \right) \right] \tag{4.15}$$

subject to:

$$\Omega(s_i) \in \{T\} \quad , \quad \forall\, s_i \in \{S\} \tag{4.16}$$

$$\Omega(s_i) \neq \Omega(s_j) \quad , \quad \forall\, s_i \neq s_j \tag{4.17}$$

$$B(l_k) \geqslant \hat{B}(l_k) \quad , \quad \forall\, l_k \in \{L(p_i)\},\ \forall p_i \in \{P\} \tag{4.18}$$

where $B(l_k)$ is the link bandwidth and $\hat{B}(l_k)$ is the link load which is determined by the application mapping function and computed as:

$$\hat{B}(l_k) = \sum_{\forall a_{i,j} \in A} b(a_{i,j}) \times \pi(l_k, p(\Omega(i), \Omega(j))) \tag{4.19}$$

where $\pi(l, p)$ is 1 if l belongs to the set of links constituting path p, $L(p)$, and 0 otherwise.

The first two constraints (Eq. 4.16 and Eq. 4.17) are used to respectively ensure that each task will be mapped to only one tile and that no more than one task can be mapped to one tile. The third constraint (4.18) is necessary to guarantee that a link load will not exceed its bandwidth capacity.

4.3.4 *Simulated Annealing-Based Solution*

The application mapping problem in NoCs is known to be NP-hard [94]. For an NoC of size $n \times m$, there are $(n \times m)!$ possible mappings. In this work a simulated annealing (SA) solution is employed. SA can help to avoid being trapped at local minima, as the temperature function would give an opportunity to escape from the minima during searching. This is suitable for NP-hard problems where suboptimal solutions are required, which is the case in this work. The pseudo code of this solution is shown in Algorithm 4.1.

The algorithm takes an application graph (APG) and an architectural graph (ARG) as inputs. Also, the final temperature $Temp_f$ and the cooling rate β need to be defined at the beginning of the algorithm. Then, both the mapping function and temperature ($Temp_c$) are set to their initial values. The parameters $Temp_c$, $Temp_f$ and β are tuned experimentally. The initial temperature, $Temp_c$ controls the level of relaxation of randomness of exploration at the start of the optimization, which needs to be chosen carefully in order to enable a good exploration of the solution space. The cooling rate β controls the speed of convergence. It needs to be chosen so as allow a sufficiently slow convergence in order to ensure a good exploration of the solution space. However, it should be not smaller than necessary since this would make the convergence too slow.

Using the initial mapping function (Ω_p), the activity densities of all tiles (γ_t, $\forall t \in \{T\}$) and the objective function (F_{tot}) are computed in lines 1-3. The main optimization loop is illustrated in lines 4-23. In lines 4-7 the new mapping (or neighbouring state) is generated by first choosing the tile with the highest activity density, and then randomly choosing another tile from its one-hop range neighbours. The tasks assigned to these tiles are then exchanged (line 7). By always modifying the tile with the highest activity, it is ensured that the next mapping would have a different hotspot. This helps in achieving a good exploration of the solution space.

The condition in line 8 checks if the new mapping satisfies the bandwidth constraint stated in Eq. 4.19. If this constraint is not satisfied, the new mapping is not valid. Thus, the new mapping is ignored and the algorithm goes directly to the next iteration. If the bandwidth constraint is met, the new activity densities and the objective function

Algorithm 4.1 Pseudo code of the force-based application mapping for activity density minimization in NoCs.

Input: -

$APG(S, A)$:Application graph,

$ARG(T, P)$:Architectural graph.

Output: -

Ω: Optimal mapping.

Define: -

$Temp_f$: Final temperature,

β: Cooling rate,

$\mathcal{N}(t)$: The set of neighbours of tile t.

Initialize: -

$Temp_c = initial\_temp$,

$\Omega = initial\_mapping$.

1: $\Omega_p \leftarrow \Omega$
2: using Ω_p, compute γ_t $\forall t \in \{T\}$
3: compute F^p_{tot} using Eq. 4.13
4: **while** $Temp_c \geqslant Temp_f$ **do**
5: $t_1 \leftarrow \arg\max_t(\gamma_t)$, $\forall t \in \{T\}$
6: choose t_2 from $\mathcal{N}(t_1)$ randomly with equal probability
7: swap the tasks of t_1 and t_2
8: **if** bandwidth constraints for Ω_p are satisfied **then**
9: Compute γ_t $\forall t \in \{T\}$ for Ω_p
10: compute F_{tot} using Eq. 4.13
11: $\Delta F = F_{tot} - F^p_{tot}$
12: **if** $(\Delta F < 0)$ **then**
13: $\Omega \leftarrow \Omega_p$
14: $F^p_{tot} \leftarrow F_{tot}$
15: **else**
16: **if** $e^{-\Delta F/Temp_c} > rand()$ **then**
17: $\Omega \leftarrow \Omega_p$
18: $F^p_{tot} \leftarrow F_{tot}$
19: **end if**
20: **end if**
21: **end if**
22: $Temp_c = \beta Temp_c$
23: **end while**
24: RETURN Ω

are computed for this new mapping (lines 9 and 10). In lines 12-20 the new mapping is accepted as the new state if it has a lower value of objective function (i.e. $\Delta F < 0$)); if not, this mapping can be accepted with a probability of ($e^{-\Delta F/Temp_c}$). Then the current temperature is reduced by the cooling rate and the loop starts again. When the current temperature is low enough, the optimization cycle ends and the resulting mapping, Ω, is returned.

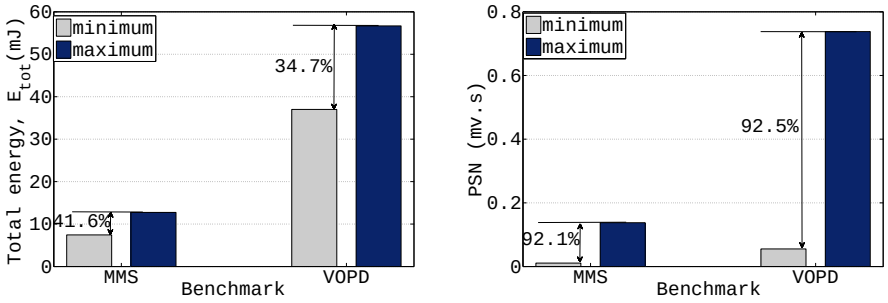
4.4 EXPERIMENTAL ANALYSIS AND RESULTS

4.4.1 *Experimental Setup*

To evaluate the proposed mapping, the PSN computation tool described in Chapter 3 is used. The PDN model includes both off-chip and on-chip power delivery networks. The on-chip PDN consists of a global level mesh structure routed in the top metal layers. A lumped model of the PDN is used and a SPICE netlist is employed in the model. For PSN computation, a 65 nm technology node is assumed. Details of the experimental setup are presented in Appendix B, Table B.1. The PSN is computed using Eq. 4.1 assuming that the noise margin, V_{NM}, is 10% of nominal V_{DD} (i.e. 100 mV), and simulations are run for 10,000 windows (10 million clock cycles).

4.4.2 *Activity Density and Power Supply Noise*

To illustrate the significance of the problem and the motivation behind the proposed PSN-aware mapping in NoCs, 100 random mappings for both the MMS and the VOPD benchmarks[137] are first analysed. For each of these mappings the resulting voltage variations were computed . The total resulting PSN (computed using Eq. 4.1) and



(a) Energy range.

(b) Power supply noise range.

Figure 4.2: Energy and PSN ranges for two benchmarks with random mapping. Different mappings can have effects on PSN. Similar results are found for other benchmarks.

the total energy E_{tot} (computed using Eq. 4.5) resulting from each

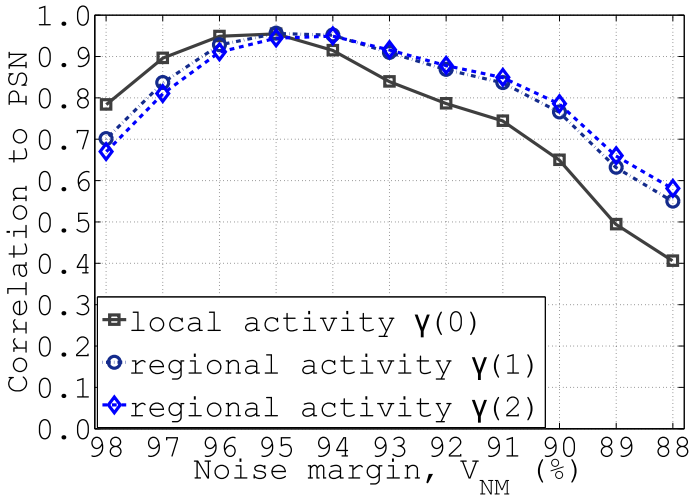


Figure 4.3: Correlation of PSN and activity density. Results from 100 random mappings for the VOPD benchmark. A significant correlation between PSN and activity density can be seen, which is higher for regional activity than local activity.

mapping are computed. The ranges of both PSN and energy for both benchmarks are shown in Fig. 4.2. It can be seen that the relative range of PSN shown in Fig. 4.2b is much higher than the range of energy (Fig. 4.2a). These results show that mapping has a strong impact on PSN which is higher than its impact on energy. This means that PSN can be reduced significantly through application mapping with low energy overheads.

Next, using the PSN results of these random mappings, the effect of spatial patterns of regional activity density on PSN is investigated. This is achieved by computing the correlation of PSN with local activity, $\gamma_t(0)$, and regional activity $\gamma_t(D)$, for two regions of sizes $D = 1$ and $D = 2$.

Fig. 4.3 plots the correlation coefficient of total PSN with both local ($D = 0$) and regional activities (for $D = 1$ and $D = 2$) against noise margin V_{NM} (see Eq. 4.1) for the VOPD benchmark. The noise margin is varied from 20 mv to 120 mv with a steps of 10 mv.

It can be noted that, for low values of V_{NM} (closer to V_{DD}), noise is highly dependent on local activity $\gamma(0)$. As the noise margin increases, noise dependence on both local and regional activities decreases. However, the correlation of the local activity ($\gamma(0)$) with PSN decreases more rapidly with V_{NM}. As a result, for higher V_{NM}, regional activity dominates local activity as a source of noise.

This change in the role of local and regional activity can be explained by the fact that, when the noise margin increases, local activity will not be sufficient to cause a voltage drop below this margin. At this point, regional activity starts to play a greater role in PSN. Also, it can be seen that regional activity for a region of size 2 ($D = 2$) has a slightly higher impact on noise than that for a region of size 1 ($D = 1$). The

same trend was found for other benchmarks. From this experimental analysis, it can be concluded that, for higher noise margins, higher region sizes need to be considered for noise optimization.

Previous work on application mapping for NoCs has considered different objectives. However, the activity density of the resulting systems is ignored. Even worse, some techniques (energy-aware mapping, for instance) usually result in higher activity density in some regions of the chip due to placing highly communicating tasks close to each other. Consequently, these regions can become hotspots, causing the chip to experience unbalanced switching activity, leading to higher PSN and reducing system reliability.

Bench. name	# of tasks	Target NoC	min./max. bw (MB/s)	$\overline{BW}$ (MB/s)	GD (%)
AMI49	49	7×7	5.3/85.3	11.60	37.07
AMI25	25	5×5	53.3/213.3	70.76	17.33
MMS	25	5×5	0.025/116.8	20.67	11.00
TELE	16	4×4	11/71	45.36	18.33
VOPD	16	4×4	16/500	177.66	17.50
MPEG4	9	3×3	8.5/502	195.12	55.55

Table 4.2: Summary of the benchmarks. GD is the graph connection density and is defined as $GD = \frac{2|A|}{|S|(|S|-1)} \times 100\%$, while the $\overline{BW}$ is the average communication bandwidth among the task graphs and is defined as $\overline{BW} = \left(\sum_{\forall a \in \{A\}} b(a)\right) / |A|$.

4.4.3 *Mapping Results*

To evaluate the proposed force-based mapping strategy in terms of PSN, six real benchmarks with different sizes, topologies and bandwidth requirements are used. These benchmarks include: a generic complex MultiMedia system which comprises an h263 video encoder and an mp3 audio decoder (MMS) [94]; a telecommunications benchmark (TELE) and Video Object Plane Decoder (VOPD) [137]; in addition to the AMI49, AMI25 and MPEG4 decoder benchmarks taken from [7]. Details of size, communication bandwidth requirements and graph density of these benchmarks are presented in Table 4.2. In this table, *graph density* (GD) is defined as the ratio of the existing connections between application tasks to the number of all possible connections. Given an application graph (APG) $G = G(S, A)$, *graph density* is defined as $GD = \frac{2|A|}{|S|(|S|-1)} \times 100\%$, while, the *average bandwidth* ($\overline{BW}$) is defined as $\overline{BW} = \frac{\sum_{\forall a \in \{A\}} b(a)}{|A|}$.

Using Algorithm 4.1, each of these benchmarks is mapped into the target NoC architecture using two strategies. The first is the proposed

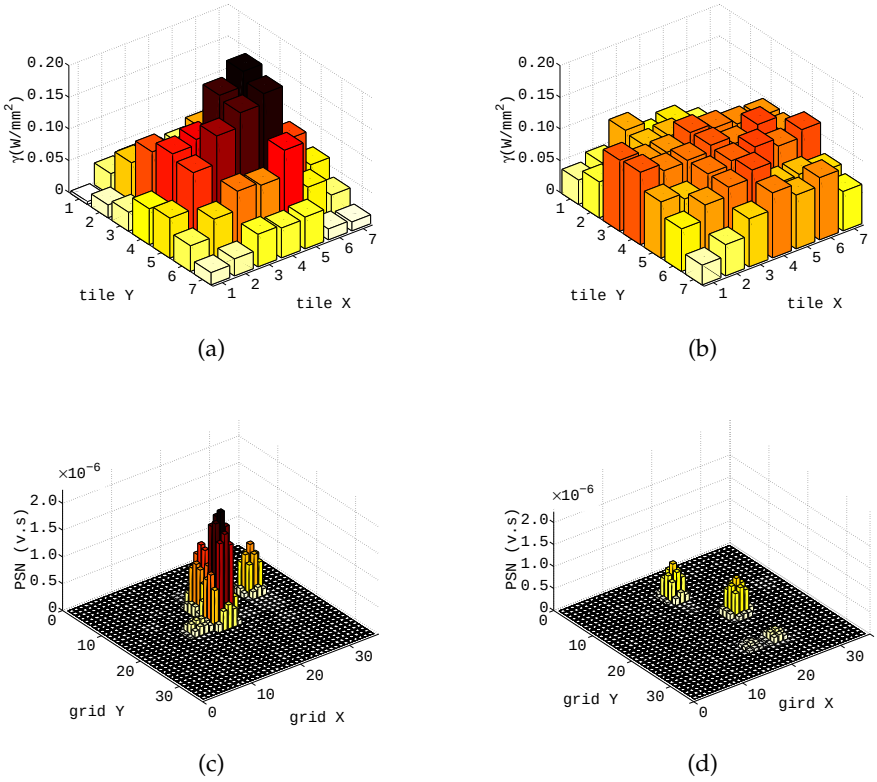


Figure 4.4: Illustration of the proposed mapping results: The spatial distribution of activity density and PSN for the AMI49 application. Significant reduction in PSN is achieved by uniform distribution of activity. a) Activity density (γ) for min. E_{tot} mapping, b) Activity density (γ) for the proposed min. F_{tot} mapping, c) PSN for min. E_{tot} mapping, d) PSN for the proposed min. F_{tot} mapping

minimum total repulsive force ($min\{F_{tot}\}$), defined in Eq. 4.13. The second strategy is minimizing the energy $min\{E_{tot}\}$ as defined in Eq. (4.5). Our noise-aware mappings are compared with energy-aware mappings in terms of both PSN and energy dissipation. Also, the performances of the two mappings are compared and discussed in Section 4.4.4.

An example depicting the impact of the proposed force mapping ($min\{F_{tot}\}$) is given in Fig. 4.4, which shows the spatial activity density and PSN distributions that result from energy and force mappings for the AMI49 benchmark.

It can be noticed that energy-aware mapping results in condensing highly active tiles, resulting in regions with high activity in the chip (see Fig. 4.4a). This results in hotspots that experience high PSN (see Fig. 4.4c). In contrast, our repulsive force minimization mapping results in a scattered and more homogeneous activity distribution (see Fig. 4.4b) which, in turn, reduces PSN significantly (by nearly 66% in this case) as depicted in Fig. 4.4d.

Bench.	$\min\{E_{tot}\}$		$\min\{F_{tot}\}$		energy	PSN
	energy (mJ)	PSN ($\mu v.s$)	energy (mJ)	PSN ($\mu v.s$)	penalty	reduction
AMI49	6.44	49.23	6.84	16.74	6.23%	65.98%
AMI25	3.21	11.97	3.33	5.33	3.72%	55.47%
MMS	5.19	2.15	5.52	1.23	6.43%	43.12%
TELE	7.90	14.31	7.98	8.57	1.01%	40.08%
VOPD	26.25	32.26	27.23	11.47	3.73%	64.44%
MPEG4	26.24	3.22	26.64	1.54	1.51%	52.23%
				Average	**3.70%**	**53.55%**

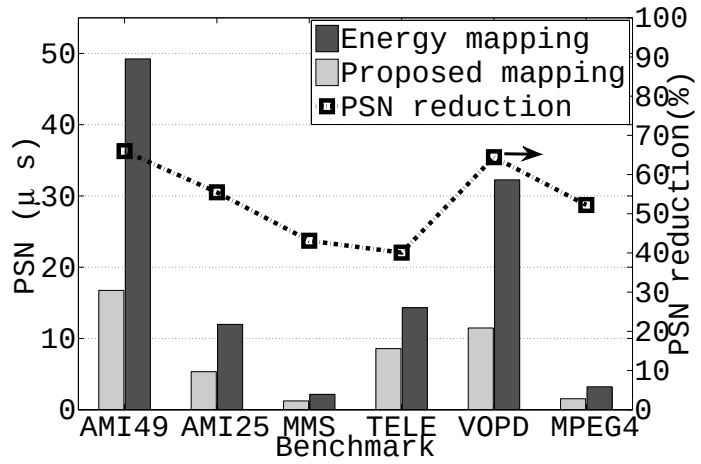
Table 4.3: Summary of mapping results in terms of the total PSN and energy consumption for both minimum energy mapping, $\min\{E_{tot}\}$, and the proposed minimum total repulsive force , $\min\{F_{tot}\}$.

The results for both PSN and energy from both mapping strategies for the six benchmarks are shown in Table 4.3. These results are also depicted graphically in Fig. 4.5. It can be seen that our repulsive force minimization results in significant drops in PSN (53.55% of noise is removed on average) with relatively low energy penalties (3.7% on average). This implies that considering regional activity in NoC mapping is very useful and can result in systems with lower noise and better power integrity.

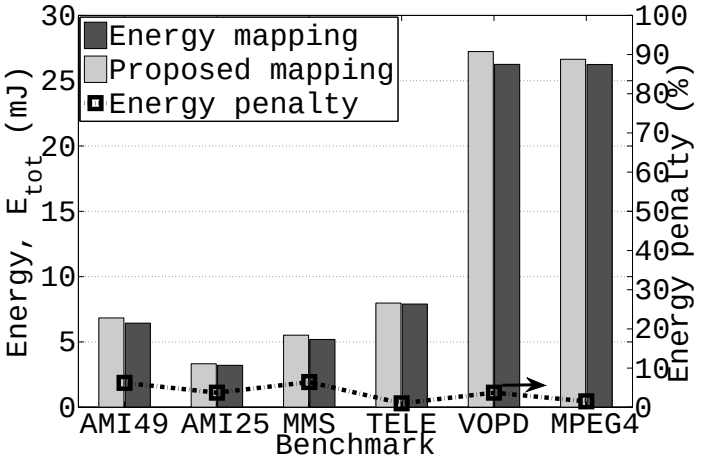
It can also be noticed that, in general, the difference in energy between energy-aware mapping and repulsive-force based mapping is relatively small. This can be explained by the fact that F_{tot}, defined in Eq. 4.13, depends on both spatial distribution and amount of energy. This explains why mappings with minimum F_{tot} have relatively low energy overhead. In other words using F_{tot} as an objective for the mapping function results in a sub-optimal solution for energy minimization. This is also verified experimentally from the very high correlations found between F_{tot} computed using Eq. 4.13 and E_{tot} computed using Eq. 4.5 for the random mappings considered in Section 4.4.2.

4.4.4 *Impact on Performance*

The impact of the proposed mapping on performance is now evaluated. This is done by computing the time required to drain 5MB of data for each application using the proposed mapping and energy-aware mapping. The results of this performance comparison are shown in Table 4.4. The proposed ($\min\{F_{tot}\}$) mapping is expected to lead to some performance degradation due to slightly higher average hop count.



(a) PSN



(b) Energy consumption.

Figure 4.5: Comparison of PSN and energy optimizations. Our noise minimization could achieve significant reductions in PSN compared to energy minimization, with a low energy penalty.

Nevertheless, it can be seen that the elapsed time for both types of mapping is very similar for all the benchmarks considered. For the majority of these benchmarks, the elapsed draining time for the proposed ($\min\{F_{tot}\}$) mapping is less than that of energy mapping. This is shown as a negative difference in Table 4.4. This can be attributed to the fact that network performance does not merely depend on the average distance travelled by the packets. Another important factor that determines network performance is the congestion status of the buffers, which affects buffer waiting time. In this sense, the proposed mapping tends to balance communication activity across the chip. Although this could lead to slightly higher average hop count, it can also lead to lower congestion and hence less buffer waiting time which can compensate for the longer path travelled by the packets. On the other hand, the $\min\{E_{tot}\}$ mapping condenses highly communicating

tasks and leads to congestion, which may cause longer packet delivery times, due to longer buffer waiting times, despite the shorter paths travelled by the packets.

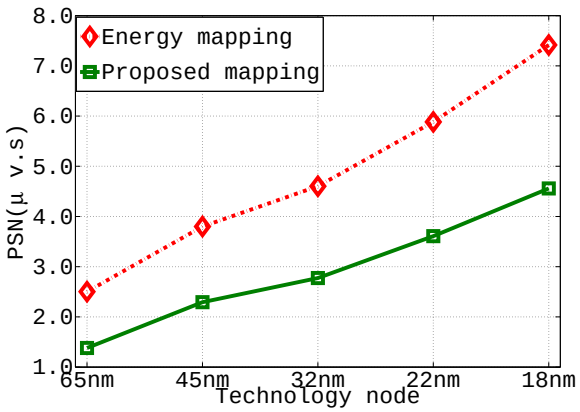
Benchmark	Draining time (µs)		Difference (%)
	energy ($\min\{E_{tot}\}$)	proposed ($\min\{F_{tot}\}$)	
AMI49	59.0	60.1	-1.86%
AMI25	90.5	89.7	0.88%
MMS	333.7	333.7	-0.01%
TELE	298.1	297.3	0.27%
VOPD	89.2	89.8	-0.66%
MPEG4	136.5	137.2	-0.47%

Table 4.4: Performance comparison of the two mappings showing time required for draining 5MB of data for each application and using the proposed mapping and energy-aware mapping strategies. Percentage difference between the two mappings are also shown.

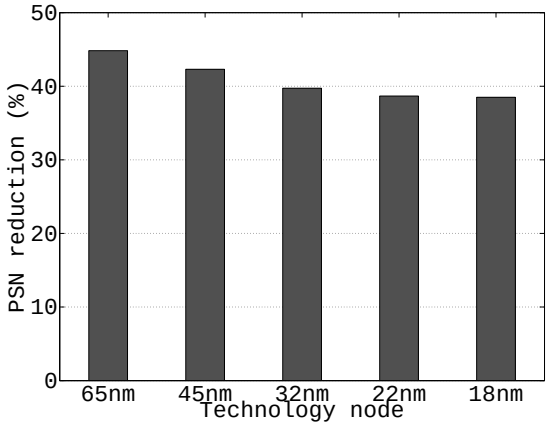
4.4.5 *Impact of Technology Scaling*

To show how the proposed mapping strategy behaves in response to technology scaling, the system is scaled down for all the considered benchmarks. Smaller technology models are obtained by scaling the nodes from 65nm down to smaller nodes. The approximate scaling of power, area, frequency and V_{DD} is achieved by scaling factors that are obtained from previous studies [103], [99] and [175]. These scaling parameters are given in Appendix B, Table B.2. Fig. 4.6 shows the resulting PSN of the MMS benchmark with different technology nodes for both mapping strategies (Fig. 4.6a). The percentage improvement with the proposed mapping relative to energy mapping is also plotted in Fig. 4.6b.

It can be seen that, in general, PSN is higher for smaller technology nodes. Moreover, it is known that for smaller technology nodes, PSN reduction is more difficult to achieve due to significantly increased energy, switching frequency, and power density [11]. Despite this, it can be noticed that, even with considerable technology scaling from 65nm to 18nm the proposed mapping strategy still achieve significant reductions in PSN compared to energy mapping. Similar results are found for other benchmarks. Fig 4.7 shows the reduction in PSN achieved by the proposed mapping for the other five benchmarks with different technology nodes. Similar results can be seen here, where the proposed mapping still achieves considerable PSN reductions even for smaller technology nodes.



(a)



(b)

Figure 4.6: Evaluation of the proposed mapping with different technology nodes: a) PSN for both the proposed ($\min\{F_{tot}\}$) mapping and energy ($\min\{E_{tot}\}$) mapping and; b) the percentage reduction in PSN, for the MMS benchmark with different technology nodes.

4.4.6 *Evaluation of Link Timing Variations and BER*

The above results show that the proposed mapping achieves significant reductions in PSN. Such reductions would translate into higher performance, less leakage and lower delay. This section evaluates the resultant mappings in terms of the timing variations that result from power supply variations, conducted by performing statistical timing analysis of the resulting mappings. This enables an estimation of rates of timing violation and reliability. This analysis is similar to the one presented in Section 3.5. However, a more accurate model, which considers the link dependency of the resulting mappings, is presented in this chapter . It was mentioned in Section 3.5 that, when considering a synchronous data path for the link between two tiles i and j, and

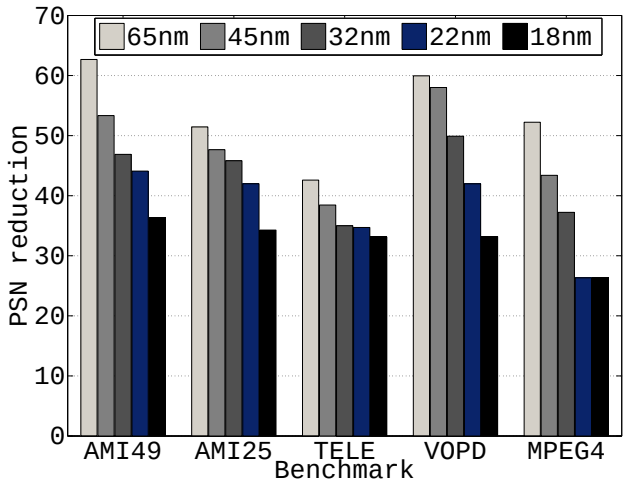


Figure 4.7: The reduction in PSN achieved by the proposed ($\min\{F_{tot}\}$) mapping compared to energy ($\min\{E_{tot}\}$) mapping for various benchmarks with different technology nodes.

assuming a zero clock skew between these tiles, the sum of all link delay components, ($t^i_{clk\_Q}$, $t^{i,j}_{wire}$ and t^j_{setup}), must be less than the clock period:

$$t^i_{clk\_Q} + t^{i,j}_{wire} + t^j_{setup} < T_{clk}. \tag{4.20}$$

Otherwise, switching errors will occur in the link. Also, a quadratic approximation is adopted to determine the impact of V_{DD} drop on these delay components:.

$$t_d(\Delta V_{DD}) = k_1 + k_2(\Delta V_{DD}) + k_3(\Delta V_{DD})^2 \tag{4.21}$$

where, $t_d(\Delta V_{DD})$ is any of the link timing components on the left hand side of Eq. 4.20 and k_i (i=1,2,3) are technology-dependent constants. Formulas for these delay components are evaluated as a function of V_{DD} using regression based on SPICE simulations. Using these formulas, the resulting delay distribution of a link can be used to estimate the probability of timing error due to power supply variations for that link. This probability is estimated as the portion of the delay distribution that violate the constraint in Eq. 4.20. In other words, the probability of timing error on link l, $P_{err}(l)$, is given by:

$$P_{err}(l) = Pr(t_l > T_{clk}) \tag{4.22}$$

where t_l is the total link delay computed in the presence of V_{DD} variations using Eq. 4.21.

Using the probability of error for each link, the average BER for the NoC can be computed. As a result of the mapping function, each

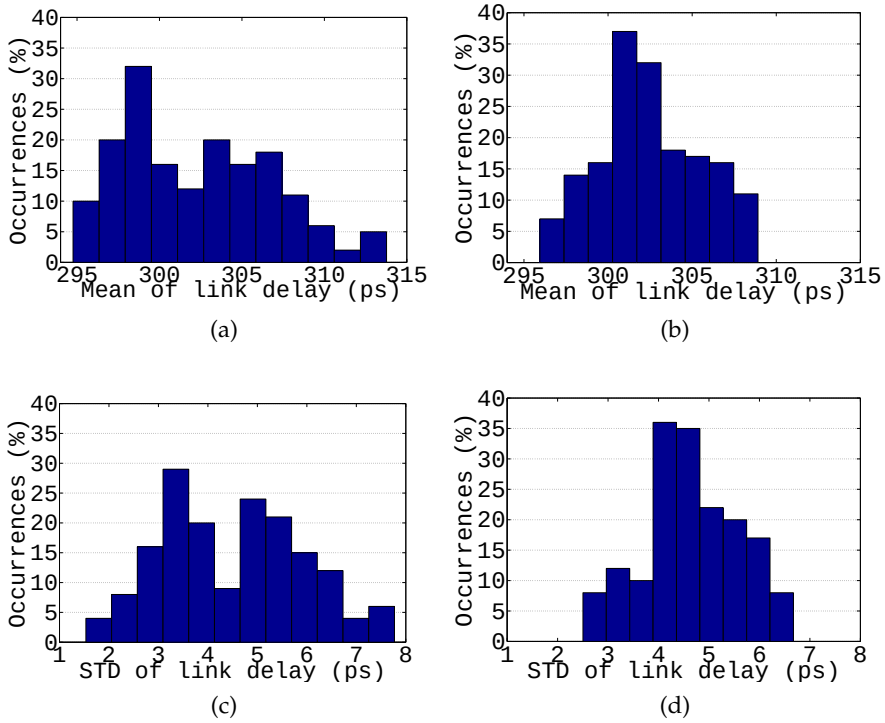


Figure 4.8: Comparison of link delay statistics for the AMI49 benchmark with both $\min\{E_{tot}\}$ and the proposed ($\min\{F_{tot}\}$) mappings. Mean of delay with; a) energy mapping; b) proposed ($\min\{F_{tot}\}$) mapping, and STD of delay with; c) energy mapping; d) proposed ($\min\{F_{tot}\}$) mapping.

arc ($a_{i,j}$) in the APG is mapped to a path ($p_{\Omega(i),\Omega(j)}$). Now, when computing the probability of error for path p, $P_{err}(p)$, link dependency along path p needs to be considered. This is because, for a particular link along the path l_i, only the data volume which does not experience errors in previous links along the path $(l_0, l_1, ...l_{l-1})$ needs to be considered. The probability of error for each link l as part of path p, $P_{err}(l,p)$ can be expressed as follows:

$$P_{err}(l_i,p) = P_{err}(l_i)(1 - \prod_{k=0}^{i-1}[P_{err}(l_k,p)]), \tag{4.23}$$

where $P_{err}(l_0,p) = P_{err}(l_0)$. In other words, for the first link in the path the probability of error for this link as a part of the path is the same as the probability of error for the link computed using Eq. (4.22).

The probability of error for the path can now be readily computed as the summation of the probabilities of its constituent links:

$$P_{err}(p) = \sum_{l\in\{p\}} P_{err}(l,p) \tag{4.24}$$

The total BER for all NoC links is the ratio of the data volume that experiences error to the total communicated data volume:

$$BER = \frac{\sum_{\forall a_{i,j} \in \{A\}} w(a_{i,j}).P_{err}(p_{\Omega(i),\Omega(j)})}{\sum_{\forall a_{i,j} \in \{A\}} w(a_{i,j})} \tag{4.25}$$

where $\{A\}$ is the set of all tasks mapped to the target NoC and Ω is the mapping function.

Fig. 4.8 shows a comparison of timing statistics for the AMI49 application with both $min\{E_{tot}\}$ and the proposed $min\{F_{tot}\}$ mappings. Figures 4.8a and 4.8b show the distribution of delay means (skews) and Figures 4.8c and 4.8d show the distribution of delay STDs (jitters) for the NoC links. It can be noticed that the proposed mapping results in lower values and ranges of both skews and jitters, thus showing better timing accuracy than energy mapping.

Table 4.5 shows the results of this timing analysis and the resulting BER for the considered applications with $min\{E_{tot}\}$ and $min\{F_{tot}\}$ mappings. It can be noticed that the reduction of power supply variations and noise achieved by our force-based mapping significantly reduces the resulting BER (up to 97% and 80.36% on average). This indicates one important advantage of our balanced activity mapping in reducing timing violations and error rates, and thereby improving reliability and performance.

Bench.	Mapping strategy		BER reduction
	$min\{E_{tot}\}$	$min\{F_{tot}\}$	
AMI49	9.67E-06	1.45E-06	84.96%
AMI25	6.11E-06	1.81E-07	97.03%
MMS	2.31E-06	8.59E-07	62.72%
TELE	2.79E-07	1.48E-07	46.82%
VOPD	1.77E-06	5.05E-08	97.14%
MPEG4	6.57E-06	4.27E-07	93.50%
		Average	**80.36%**

Table 4.5: The resulting BER reduction achieved by the proposed mapping strategy compared to energy mapping for various real benchmarks.

4.5 SUMMARY AND CONCLUSION

PSN has a significant impact on system reliability. In particular, on-chip communication could easily incur PSN without proper isolation. This chapter proposes a new mapping strategy which aims to reduce PSN in inter-core communication through the optimization of activity distribution. It is found that different mappings can result in a significant variations in PSN (up to 92.5%). A new metric based on communication

activity density, which has a direct impact on PSN, has been developed. This new metric is integrated into a new mapping algorithm and a repulsive force-based strategy is employed. This new strategy leads to a balanced distribution of activities across the chip. As a result, the new mapping strategy can achieve significant PSN reductions (up to 66%) with negligible energy and performance penalties. Also, the PSN reduction achieved by the proposed mapping is shown to be consistent for smaller technology nodes. Moreover, a statistical timing analysis of NoC links show that this reduction in PSN is reflected in significant reductions in BER (up to 97%), enabling better power supply integrity and reliability for future many-core systems.

5 DYNAMIC THERMAL OPTIMIZATION IN 3D NOCS

5.1 INTRODUCTION AND MOTIVATION

Semiconductor manufacturing processes are approaching the physical limits. This motivated the exploration of the use of 3D VLSI design, which could have many advantages including shorter global interconnect lengths, less delay, better scalability and smaller form factors. On the other hand, the NoC has been proposed as a promising communication paradigm for SoC and CMP systems which could overcome the limitations associated with on-chip bus connectivity. NoCs provide a scalable, flexible and power efficient solution to integrate many cores in a single chip [23, 63].

In 3D NoCs benefits can be gained from both 3D integration and NoCs [199, 204, 12, 167], with shorter interconnects, smaller form factors and reduced delay leading to major performance increase in 3D SoCs and 3D CMPs compared to 2D systems [71, 200]. However, future 3D VLSI systems in general, and 3D NoCs in particular, are prone to thermal challenges due to decreased transistors junction temperature, exacerbated spatial temperature gradients and increased device density. For these reasons, worst-case cooling system design will not be feasible. Instead, run-time thermal management (RTM) techniques at various levels of optimization are indispensable, especially at the network level where the NoC communication power budget takes up a significant portion of overall chip power, and may dominate logic as a source of heat [170]. NoCs power consumption would increase in the future due to advances in both technology and architecture.

Technology scaling is causing interconnects to consume more power than logic [104]. This is mainly because smaller technology reduces delay of logic gates and their power consumption, but results in relatively slower and more power-hungry wires. This is due to the fact that wires do not scale in the same way as logic. It is expected that in future technology nodes, interconnect power would take up to 65%-80% of total chip power [159].

Meanwhile, in terms of architecture, an important trend in the microarchitecture of many-core systems advocates integrating many (hundreds or thousands of) simple cores rather than integrating few complex cores. This has many advantages, such as the higher performance and finer control of these simple cores [33]. As a result, the complexity of the individual core is decreasing their number increases, causing the communication power budget to increase relative to computation power.

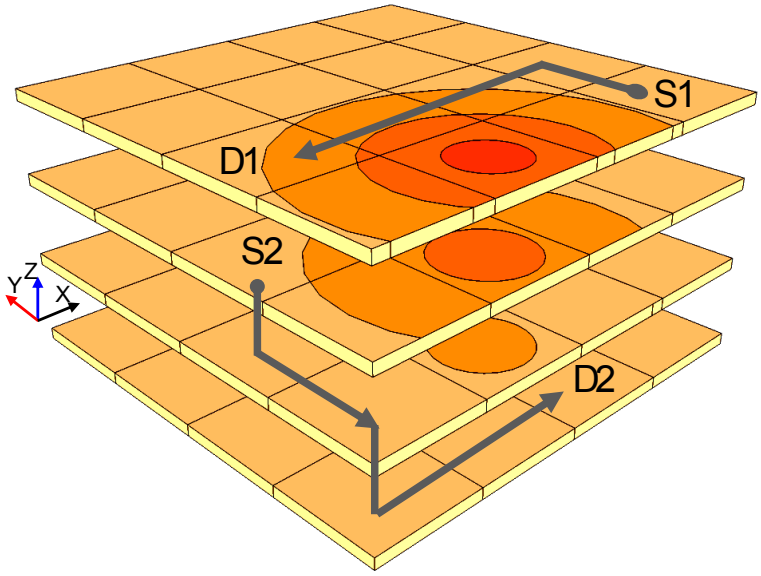


Figure 5.1: Illustration of thermal-aware routing paths.

As a result of the aforementioned reasons, future many-core CMP design is becoming communication-centric, causing NoCs to contribute significantly to power consumption and chip heating. Even in present designs, the NoC is shown to make a contribution to heat generation either comparable or greater than that of processors, particularly for communication-intensive applications. Examples are the MIT-RAW chip [170] and Intel single chip cloud computer [161]. Furthermore, for Intel's 80-core CMP, the power density of the NoC routers is nearly double that of other units such as the floating-point and memory units in the tile [193]. Thus, the NoC would make a greater contribution to chip heating compared to these units.

Moreover, controlling the NoCs workload offers a unique opportunity to control a workload spanning the whole chip. This implies that the thermal-aware control of routing paths could achieve better thermal distribution over the entire chip. In this chapter, an adaptive RTM scheme for 3D NoCs is proposed. This scheme is based on a routing strategy which tries to reduce thermal hotspots by effectively migrating routing load towards the coolest regions in the 3D geometry in order to achieve thermal optimization. Fig. 5.1 gives an example depicting how the proposed routing strategy works. In this figure, there are several shortest paths between the source node S1 and the destination node D1. However, paths that involve thermal hotspots are avoided and the proposed scheme chooses the coolest path among those available to reduce the power density at these nodes and thus moderate the temperature. Similarly, in another example for inter-layer paths between source node S2 and destination node D2, a path which avoids thermal hotspots is selected. The contributions of this chapter can be summarized as follows:

1. A new runtime thermal-adaptive routing strategy is proposed to effectively diffuse heat from the NoC-based 3D CMPs, and a distributed dynamic programming (DP)-based control architecture, dynamic programming network (DPN), is introduced to implement the proposed strategy.

2. The DPN is improved such that costs are computed and propagated in compliance with deadlock-free routing algorithms. Furthermore, 3D deadlock-free adaptive routing algorithms are improved in order to achieve higher path diversity, more balanced adaptiveness and better performance.

3. The proposed methodology is evaluated through experimental studies and comparisons with state-of-the-art NoCs RTM techniques using various synthetic traffic scenarios and real benchmarks. Results for temperature, reliability, energy and performance are compared and discussed.

4. The hardware implementation of the proposed method is discussed in detail and area and power overheads are evaluated.

5.2 RELATED WORK

Network-on-chip design space exploration may involve many design-time and runtime optimization techniques [130]. Dynamic and static application mapping and scheduling which aim to maximize performance or minimize energy are examples of these techniques [203, 93]. When the application mapping and traffic characteristics are known, the communication paradigm can be optimized to meet various objectives. The key to this optimization is designing a static or dynamic routing algorithm which determines the path taken by data packets.

The routing algorithm is important as it impacts on all network metrics such as latency, throughput and power consumption. It can also affect thermal issues, since temperature distribution is highly correlated with power consumption. This is particularly so when the NoC takes up a substantial portion of system power, e.g. 40% in the MIT RAW chip [181] and 30% in the Intel 80-core teraflop chip [192].

On the other hand, due to the greater thermal challenge in current and future VLSI systems, thermal modelling and management has gained a lot of attention in recent years [172, 65, 122, 38]. For example a generic modelling of the thermal behaviour of VLSI chip has been proposed [172]. This model starts from basic a RC dynamic compact model in modelling the main heat transfer path with typical package settings. A thermal modelling tool called *HotSpot* has also been released [96].

Thermal management methods have also been studied by different researchers. For example, distributed task migration is proposed for

thermal management [75]. This strategy relies on distributed agents to proactively exchange tasks among neighbouring cores in order to balance the workload and avoid thermal emergency. In another work, DVFS is used to avoid exceeding the emergency temperature [65]. On-line task allocation to reduce hotspots and avoid exceeding the thermal limit has also been proposed [126].

Some runtime thermal management schemes for on-chip networks that mainly uses traffic throttling to avoid thermal emergencies have been proposed [170, 169, 122]. However, exploiting NoC routing to better control chip heat distribution has received limited attention in existing literature. A characterization of the thermal profile in the MIT Raw chip [169] revealed that the NoC can surpass processors in heat generation for highly parallelizable and communication-centric applications. For example, for the 802.11a encoding application, the chip temperature reached 53°C when only processors workload was considered, while with NoC workload alone the chip temperature reached 60°C. Similar proportions were found for the 8b_10b encoding and the FIR applications [169]. Motivated by these results, a routing-based NoC RTM strategy has also been proposed. In the proactive phase of this strategy, neighbouring nodes exchange traffic counters as a mean of thermal balancing. When the thermal limit of the chip is violated, throttling is proposed as a reactive strategy [170]. In another work [157], thermal-aware application-specific routing path allocation was proposed for 2D mesh MPSoCs. The authors propose using linear programming (LP) to allocate routing paths at design time such that thermal variations of among the cores are compensated and thermal hotspots are minimized. However, this scheme is an offline technique and cannot adapt to application dynamics at runtime.

In 3D NoCs, significant thermal variations among the 3D layers can occur due to longer heat paths to the heat sink. In a recent study [42] the authors proposed a non-minimal routing called downward XYZ routing (dw_xyz) for 3D NoCs to migrate routing load, and thus power consumption, to layers closer to the heat sink. This would improve the efficiency of heat diffusion. The downward level is determined by traffic counters in order to minimize the impact of downward routing on performance and to avoid congestion. However, the implementation of this approach requires a H/W overhead for holding, updating and communicating these counters. Other shortcomings of this scheme are that cool paths within the layer are not exploited, and furthermore it is tailored for a particular cooling system and cannot adapt to different cooling systems.

The present work attempts to provide a routing scheme which is flexible in manoeuvring packets away from hot paths. The proposed routing exploits cool paths wherever they are and whenever they become available in the chip, using a DP-based distributed control architecture.

5.3 PROBLEM DEFINITION AND BACKGROUND

This section presents definitions and the necessary background for this chapter. This includes a discussion of the motivation behind RTM techniques and an introduction to temperature-related failure mechanisms.

5.3.1 *Thermal Optimization and Management*

Due to continuous shrinking of feature size, severe thermal challenges emerge. Thus, design-time thermal optimization is becoming increasingly difficult. Furthermore, 3D die stacking results in higher spatial temperature gradients over different strata due to longer main heat flow paths. Thus, worst-case cooling system designs are becoming infeasible due to the prohibitive packaging cost associated with such designs [170]. Alternatively, RTM techniques can be used. These techniques would diffuse heat and regulate the system's operating temperature at runtime before the thermal limit is exceeded, in order to keep it within a safe range. In this scenario, chip and package design for typical cases would be possible.

Techniques that use RTM monitor the temperature at runtime and alter system behaviour accordingly. These techniques can be divided into two categories: reactive and proactive. Reactive techniques work when the thermal limit is exceeded and sacrifice performance in order to achieve thermal regulation (e.g. DVFS). On the other hand, proactive techniques try to reduce thermal hotspots and minimize the temperature at runtime. This reduces, and may alleviate, the need for reactive action, thus improving both chip performance and reliability. Examples of these techniques are dynamic task scheduling and allocation in CMPs [50, 126].

5.3.2 *Temperature-Related Faults*

Higher temperatures can lead to slower devices and increases leakage current as well as interconnect delay due to higher resistivity. Furthermore, a higher thermal gradient over different chip regions may lead to failure of timing closure and increase in soft errors. However, the most prominent impact of higher temperatures and thermal gradients is on lifetime reliability [35]. It has been reported that a 10°C increase in chip temperature would cause the lifespan of the device to be shortened by half [168]. As a result, increased temperature and spatial thermal variation (or temperature gradients) increases the mean-time-to-failure (MTTF), reducing chip reliability and shortening the device lifespan. Moreover, it has been reported that over 50% of electronic products failures are temperature-related [170, 108].

The impact of a failure mechanism is usually expressed in terms of the MTTF. Failure mechanisms include the following [108, 35, 174]:

1. **Electromigration** failure is caused by the displacement of interconnect mass due to the flow of electrical current. This will lead to thinner wires and higher resistance in interconnects. Eventually, it can result in interconnect faults due to an open circuit. The MTTF for electromigration is given by [108]:

$$MTTF_{EM} = \frac{A_{EM}}{J^n} e^{\frac{E_{aEM}}{kT}} \tag{5.1}$$

where T is temperature, A_{EM} is a constant, J is current density, k is Boltzmann's constant, E_{aEM} is the activation energy of electromigration, and n is an empirically determined constant.

2. **Time-dependent dielectric breakdown** is caused by the breakdown of the gate oxide dielectric, which results in a conductive path in this dielectric. This failure becomes more prominent with technology scaling due to lower dielectric thickness, lower operating voltages and higher operating temperatures. The MTTF due to time-dependent dielectric breakdown is given by [108]:

$$MTTF_{TDDB} = A_{TDDB} \left(\frac{1}{V}\right)^{(a-bT)} e^{\frac{A+B/T+CT}{kT}} \tag{5.2}$$

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

3. **Stress migration** is similar to electromigration but is caused by the migration of interconnect mass atoms due to mechanical stress caused by mismatches in thermal expansion for different materials. The MTTF due to stress migration is given by [108]:

$$MTTF_{SM} = A_{SM} |T_0 - T|^{-n} e^{\frac{E_{aSM}}{kT}} \tag{5.3}$$

where A_{SM} is a constant, T_0 is the metal deposition temperature during fabrication, n is an empirically determined constant, and E_{aSM} is the activation energy of stress migration.

4. **Thermal cycling** is IC fatigue failure which accumulates every time there is a cycle in temperature. It is also caused by thermal expansion mismatch and occurs mainly in adjacent die and package metal layers (e.g. solder joints). The MTTF due to thermal cycling is given by the Coffin-Manson equation as follows [108]:

$$N_f = C_o \frac{1}{(T_{average} - T_{ambient})^q} \tag{5.4}$$

where N_f is the number of cycles to failure, C_o is a constant, $T_{average}$ is the average chip temperature, $T_{ambient}$ is the ambi-

ent temperature, and q is the Coffin-Manson exponent constant. Now MTTF due to thermal cycling can be expressed as the product of N_f and the average time of a thermal cycle, t_{TC}, as follows:

$$MTTF_{TC} = A_{TC} \frac{1}{(T_{average} - T_{ambient})^q} \tag{5.5}$$

where, A_{TC}=t_{TC}.C_o.

To calculate the total MTTF, the effects of various failure mechanisms need to be combined. A model such as the sum-of-failure-rates model (SOFR) can be used to obtain the resultant failure rate due to different fault mechanisms. This model is based on two assumptions: (1) system failure is a series of failures, where any failure due to any mechanism will cause the entire system to fail; and (2) all failure mechanisms have constant failure rates. Under these assumptions, the total failure rate (λ_{tot}) can be expressed as:

$$\lambda_{tot} = \frac{1}{MTTF_{tot}} = \sum_{\forall f \in \{F\}} \sum_{\forall k \in \{K\}} \lambda_{fk} \tag{5.6}$$

where {F} is the set of faults, {K} is the set of components in the system, and λ_{fk} is the fault rate of component k due to fault f. The fault rate is the reciprocal of MTTF.

The conventional way of expressing failure rates for electronic devices is in terms of failures-in-time (FIT). The FIT value is the number of failures expected in one billion (10^9) device-hours. Thus, the total FIT can be expressed as:

$$FIT_{tot} = \lambda_{tot} \times 10^9 \tag{5.7}$$

FIT is used to report improvements in reliability as a result of temperature reductions achieved by the thermal optimization methods discussed in this chapter.

5.4 DPN-BASED THERMAL OPTIMIZATION IN 3D NOCS

In this section the proposed DPN-based RTM is presented. All aspects associated with the DPN are discussed such as dynamic programming, routing algorithm deadlock-freeness, routing algorithm adaptiveness and convergence time.

5.4.1 *Shortest Path Computation using Dynamic Programming*

Dynamic programming (DP) is an efficient optimization method that is suitable for problems that can be broken down into subproblems. Problems which can be solved with decisions that span several points

in time recursively, and where *Bellman's principle of optimality* can be applied, are said to have optimal substructures and can be solved using DP [49].

One such problem is the *shortest path* problem in graph theory. This problem is essential in NoCs runtime management. Runtime dynamic routing with congestion avoidance, fault tolerance or thermal management can be formulated as shortest path problems [129]. Figure 5.2 illustrates the shortest path as a bold line between a source node (S) and a destination node (D).

The shortest path problem can be described as follows: Given a directed graph $G = (\mathcal{V}, \mathcal{A})$ with $N = |\mathcal{V}|$ nodes, $m = |\mathcal{A}|$ edges, and a cost $C_{u,v}$ associated with each edge $u, v \in \mathcal{A}$. The total cost of a path of length l, $p = \langle n_0, n_1, \ldots, n_{l-1} \rangle$ is the sum of the costs of its constituent edges: $Cost(p) = \sum_{i=1}^{l-1} C_{i-1,i}$. The shortest path from a source node s to a destination node d is then defined as any path p with minimum cost, $min\{Cost(p)\}$, $\forall p \in P_{s,d}^{l}$, where $P_{s,d}^{l}$ is the set of all paths between s and d.

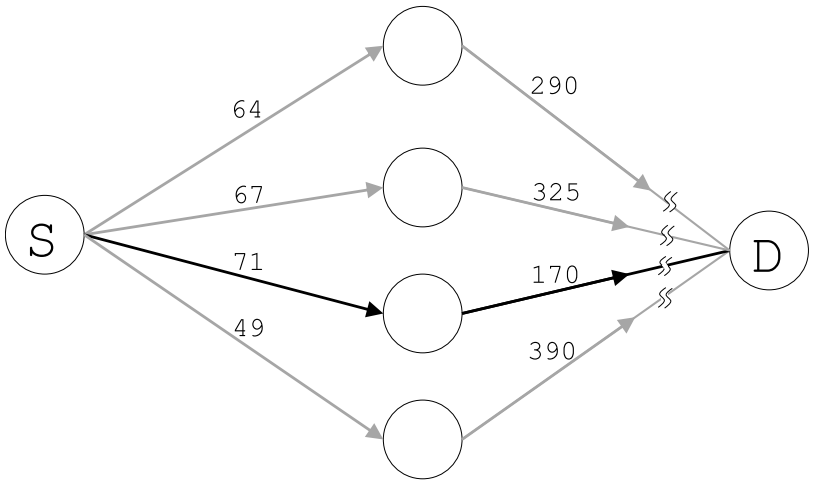


Figure 5.2: Illustration of finding the shortest path in a graph: a straight line indicates a single edge; a discontinuous line indicates a shortest path between the two nodes it connects (other nodes on these paths are not shown); the bold line is the overall shortest path from source, S, to destination, D.

The shortest path problem can be readily formulated and solved using a standard linear programming solver. Alternatively, it can be simplified by breaking it down into simpler subproblems and then solved recursively using DP [90]. Solving the shortest path problem using DP involves stating this problem in the form of Bellman equations, which define a recursive procedure in step k and can lead to a simple parallel architecture to speed up the computation.

In DP, finding the shortest path from a source node s to a destination node d requires computing the DP value or namely the *cost-to-go*, which is the expected cost from s to d. This cost is updated recursively until it reaches its optimal value. The DP value from s to d at the k-th

iteration is denoted as $V^{(k)}(s,d)$ here. After the algorithm converges the optimal DP value, $V^*(s,d)$, will hold the minimum cost from s to d. For any intermediate node u between s and d, the Bellman equation can be written as follows:

$$V^{(k)}(s,d) = \min_{\forall u \in \mathcal{V}} \left\{ V^{(k-1)}(u,d) + C_{s,u} \right\} \tag{5.8}$$

Starting from $u = d$ and $V(d,d) = 0$, the recursion can be expanded for a path of length l between nodes s and d, i.e. $s = n_0$ to $d = n_{l-1}$. The optimal DP-value can then be expressed as the total cost of the optimal path from node s to node d:

$$V^*(s,d) = \min_{p \in P^l_{s,d}} \{Cost(p)\} \tag{5.9}$$

where $V^*(s,d)$ is the optimal (minimum) cost for the path between s to d and $P^l_{s,d}$ is the set of all paths of length l from s to d.

From this minimum cost path (or shortest path), the optimal decision (direction) to the destination node can be readily obtained from the argument of the minimum operator, as follows:

$$\mu(s,d) = arg \min_{\forall u \in \mathcal{V}} \{V^*(u,d) + C_{s,u}\} \tag{5.10}$$

where $\mu(s,d)$ is the optimal decision, or direction, to be taken in order to reach destination node d with the minimum cost.

Cost can be associated with nodes rather than edges. This is the case in this chapter, since the cost is defined as the router temperature. In such cases, the costs of all directed edges entering a node are equal to the cost associated with this node, which is the router's local temperature (T_{local}).

5.4.2 *DPN Guided 3D NoC Routing*

The proposed routing strategy relies on distributed DP units to guide the routing load to the coolest path in the chip. These units are connected via a dynamic programming network (DPN).

The DPN is tightly coupled with the NoC communication fabric, and consists of distributed computational units. At each NoC router, there is a DP unit which implements the DP algorithm and propagates the solution to the neighbouring units. Each computational unit locally exchanges control and system parameters with the corresponding NoC router.

Assuming a multi-source single destination, each DP unit receives the cost of the neighbouring units as input, and computes and propagates the minimum cost to the neighbours after adding its local temperature. The temperature is used as the node cost and, thus, the shortest path from a source to a destination is the one with minimum

Algorithm 5.1 Operations performed by the DP-unit for thermal optimization.

Define: -

n_c: Current node,
$V'(n_c, k, n_d)$: Cost of sending packet from n_c to n_d through channel k,
$\mathcal{N}(n_c)$: All neighbour nodes of node n_c,
$\mathrm{ROUTE}(n_c, n_d)$: Routing function that takes n_c and n_d and return candidate routing directions $K_{n_c,n_d} \subset \mathcal{N}(n_c)$,
par: denotes parallel operations.

Inputs: -

n_d: Destination node,
$V(i, n_d) : \forall i \in \mathcal{N}(n_c)$, Costs for all neighbours of n_c to n_d,
T_{local} : Router local temperature.

Outputs: -

$\mu(n_c, n_d)$: Optimal direction from node n_c to n_d,
$V^*(n_c, n_d)$: Optimal cost from node n_c to n_d.

1: **if** $n_d = n_c$ **then**
2: $V^*(n_c, n_d) = 0$
3: $\mu(n_c, n_d) = \mathrm{LOCAL}$
4: **else**
5: $\mathrm{Cost}_c = T_{local}$
6: $K_{n_c,n_d} = \mathrm{ROUTE}(n_c, n_d)$
7: **par for all** directions $k \in K_{n_c,n_d}$ **do**
8: $V'(n_c, k, n_d) = V(k, n_d) + \mathrm{Cost}_c$
9: **end par for all**
10: $V^*(n_c, n_d) = \min_{\forall k} V'(n_c, k, n_d)$
11: $\mu(n_c, n_d) = \arg\min_{\forall k} V'(n_c, k, n_d)$ {Update optimal directions}
12: **end if**

total temperature (i.e. the coolest). In this scenario, thermal hotspots are avoided whenever possible and the coolest paths are always exploited to minimize the thermal effect of the routing workload.

Considering the costs of all neighbours in the DP unit's decision implies that packets can be relayed in any direction towards the destination. However, this is only possible for fully-adaptive routing, which cannot guarantee deadlock-freeness. One alternative is to use partially-adaptive routing, which guarantees deadlock-freeness by prohibiting some turns in order to break waiting cycles. Thus, cost propagation and computation must only consider *possible* directions, which are those allowed by the routing algorithm. Algorithm 5.1 presents the operations required for updating the routing directions using the DP unit. The router's local temperature, which is used as the cost in the DP unit computation, comes from the distributed embedded sensor in the chip. The main algorithm is outlined in lines $1-12$. If the current node is the destination, the DP-unit outputs zero as a cost and the routing decision is the local port (lines $1-4$). For other

destinations, the local cost is computed as shown in line 5. Each DP unit takes the optimal cost of each neighbour node as input. However, as mentioned earlier, the DP unit should consider only candidate neighbours returned by the routing function ROUTE as shown in line 6.

Given a destination n_d and a direction k, the expected cost is computed for all routable directions (lines 7-9). Then the minimum cost is selected in line 10. The optimal routing direction is selected and used to update the routing directions in line 11. The outputs of the unit at node n_c, for a given destination n_d, are the updated expected cost $V^*(n_c, n_d)$ and the best direction $\mu(n_c, n_d)$. $V^*(n_c, n_d)$ is propagated to all neighbouring nodes (to perform a similar operation), while $\mu(n_c, n_d)$ is sent to the local router to update the routing table.

Although Algorithm 5.1 has a loop, it can be performed in hardware using parallel architecture and the computational delay reduces to linear. Computational delay in the DP unit and its convergence are discussed in Section 5.4.5, while the hardware realization of the DP unit is detailed in Section 5.6.6.

5.4.3 *Deadlock-Freeness and Adaptiveness*

In Algorithm 5.1 the degree of adaptiveness offered by the routing function ROUTE plays a crucial role in DPN performance. For instance, if the routing is deterministic XYZ, the DPN has no impact on routing paths. Conversely, for fully-adaptive routing, the routing paths is completely determined by the DPN. However, fully-adaptive routing is prone to deadlocks and requires deadlock detection and recovery techniques [121]. These techniques have their power and area overhead and would impact performance. Thus, in this work, an turn model is improved and adopted to ensure deadlock-freeness of the proposed routing.

5.4.3.1 *Balanced 3D Routing*

In this work, an improved 3D turn model is used to ensure the deadlock-freeness of the proposed adaptive routing. The proposed 3D turn model is based on a 2D *odd-even* turn model. The odd-even routing is considered to be an improvement in terms of its degree of adaptiveness compared to other turn model routing algorithms [44]. The odd-even routing differs from other turn models in prohibiting different turns for odd and even columns in a 2D.

To describe the proposed 3D routing, the rules of odd-even in 2D meshes as first summarized as follows [44]:

- **Rule 1**: *In odd columns packets are allowed to take neither North-West (NW) nor South-West (SW) turns,*

- **Rule 2**: *In even column packets are allowed to take neither East-North (EN) nor East-South (ES) turns.*

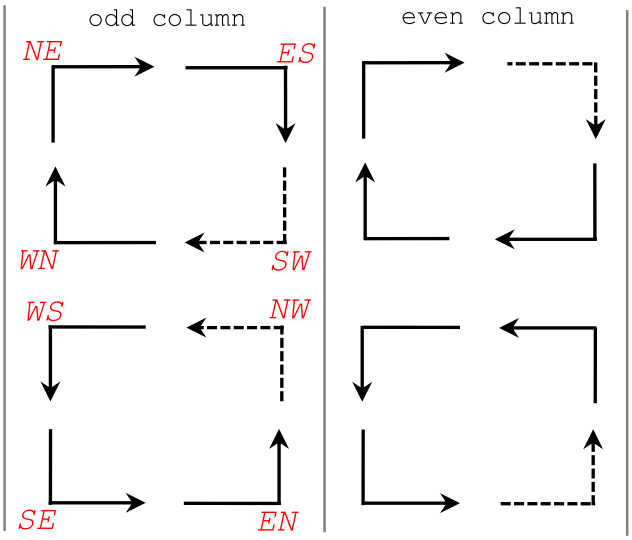


Figure 5.3: Illustration of prohibited turns for odd-even routing (rules 1 and 2). Dashed lines represent prohibited turns.

The deadlock-freeness of the odd-even turn model is proven by contradiction [44]. These rules are illustrated in Fig. 5.3. For a waiting cycle to exist in a 2D mesh NoC, both ES and SW have to occur in the same column (for clockwise cycles), or both EN and NW turns have to occur in the same column (for counter-clockwise cycles). Both scenarios contradict rules 1 and 2, since these rules ensure that the column of an NW turn cannot have an EN turn and the column of a SW turn cannot have an ES turn. Thus, the odd-even turn model defined by rules 1 and 2 is deadlock-free.

Extending the 2D odd-even model (described by rules 1 and 2) to 3D meshes requires the application of a rule to ensure deadlock-freeness and prohibit waiting cycles that consists of vertical turns (turns involving Up or Down directions). The following rule ensures this:

- **Rule 3**: xy − Down *turns are not allowed in an odd* xy*-plane and* Up − xy *turns are not allowed in an even* xy*-plane.*

In other words, packets travelling upward cannot enter an even xy-plane (turn North, East, South or West) and packets travelling within an odd xy-plane cannot leave this plane in the downward direction (as illustrated in Fig. 5.4). The 3D odd-even routing that is described by rules 1, 2 and 3 can then be called *conventional odd-even* or, for short, oe.

Other versions of 2D odd-even routing can be defined. Here a modified odd-even routing is defined with turn prohibitions that are applied according to the row, and not the column, of a 2D mesh. The rules of the modified odd-even can be stated as follows:

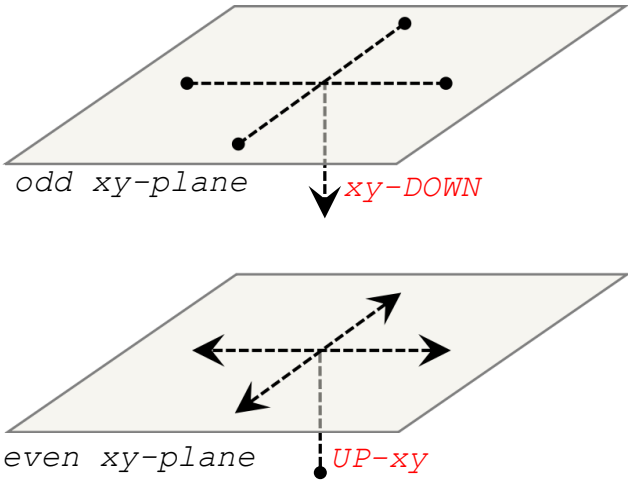


Figure 5.4: Illustration of prohibited vertical turns for the 3D odd-even routing (rule 3).

- **Rule 4**: *In odd row packets are allowed to take neither West-North (WN) nor East-North (EN) turns,*
- **Rule 5**: *In even row packets are allowed to take neither South-West (SW) nor South-East (SE) turns.*

Similar to the conventional odd-even turn model defined by rules 1 and 2, the odd-even turn model defined by rules 4 and 5 is deadlock-free and the proof is similar to that of the conventional odd-even. Looking at the waiting cycles row-wise instead of column-wise, the row of the SW turn cannot have a WN turn (prohibiting any clockwise cycles) and the row of the SE turn cannot have an EN turn (prohibiting any counter clockwise cycles). Thus, a 2D NoC that applies rules 4 and 5 is deadlock-free, since no waiting cycles can exist.

The proposed extension of 2D partially adaptive routing to 3D NoCs is based on the following corollary.

Corollary 1. In 3D NoCs deadlock-freeness is still guaranteed when different layers have different turn prohibition rules if these rules guarantee intra-layer deadlock-freeness and a rule is applied to guarantee freeness from deadlocks that involve vertical turns.

Proof. Corollary 1 is proven by contradiction. Assume that there are a set of packets that form a deadlock cycle in a 3D mesh. Then this cycle must be either on the same plane (planar deadlocks) or span two or more planes (3D deadlocks). The first case contradicts Corollary 1 since it states that intra-layer rules must exist to guarantee deadlock-freeness within the layer. The second case cannot occur since vertical links cannot be part of a cycle if a rule is applied to guarantee freeness from deadlocks that involve vertical turns as stated by the corollary. Thus, any 3D routing algorithm that satisfies Corollary 1 is deadlock-free. □

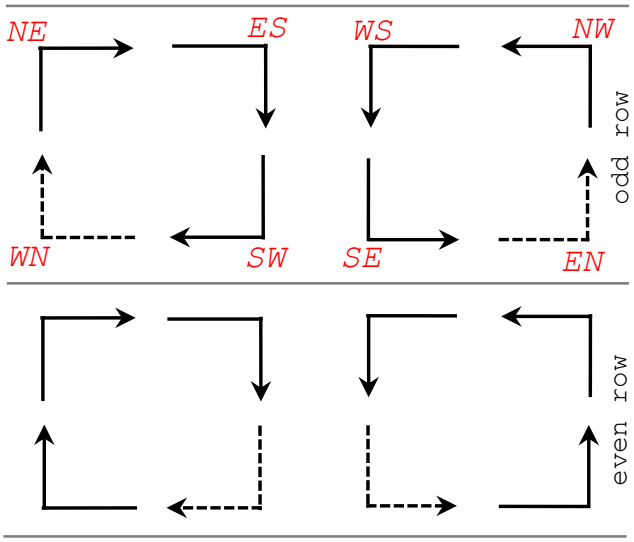


Figure 5.5: Illustration of prohibited turns in modified odd-even routing (rules 4 and 5).

For conventional 3D odd-even routing, odd-even rules within the xy -plane are the same for all planes. They are applied along the *column*. Based on Corollary 1, different rules can be used for different layers to achieve more balanced adaptiveness. Thus, the conventional 3D odd-even can be modified such that the rules for the odd xy-plane (where the z coordinate is odd) are different from those for the even xy -plane (where the z coordinate is even). The 3D routing algorithm proposed in this work *uses rules 1 and 2 in an odd plane, and rules 4 and 5 in an even plane*. These rules are applied to prohibit planar deadlocks. Rule 3 is used to prohibit 3D deadlocks. The resulting 3D odd-even algorithm balances the adaptiveness among different planes in a 3D mesh, as will be seen below, and is called *balanced odd-even* (boe). Both oe and boe are deadlock-free since they satisfy Corollary 1.

5.4.3.2 *Degree of Adaptiveness*

The degree of adaptiveness is one of the metrics that is used to evaluate adaptive routing algorithms [79]. It can be defined as the number of different allowable paths from a source to a destination. For a 3D mesh, let the coordinates of the source node be (x_s, y_s, z_s) and the coordinates of destination node are (x_d, y_d, z_d). Also, in the following, let $d_x = |x_d - x_s|$, $d_y = |y_d - y_s|$ and $d_z = |z_d - z_s|$.

For fully adaptive routing, the degree of adaptiveness is the number of *all* shortest paths from source to destination and is given by:

$$P_{fully\_adaptive} = \frac{(d_x + d_y + d_z)!}{d_x! d_y! d_z!} \tag{5.11}$$

As a result of applying Rules 1 and 2 in all planes, the degree of adaptiveness of the conventional oe (P_{oe}) can be expressed as follows:

$$P_{oe} = \frac{(h + d_y + k)!}{h! d_y! k!} \tag{5.12}$$

where h is equal to $\lceil \frac{d_x}{2} \rceil$ or $\lceil \frac{d_x - 1}{2} \rceil$ depending on the column at which x_s lies and d_x. Similarly, k is equal to $\lceil \frac{d_z}{2} \rceil$ or $\lceil \frac{d_z - 1}{2} \rceil$ depending on the layer at which z_s lies and d_z. It can be noted that, for the conventional oe, the constrained directions are x and z while the y direction is relaxed. On the other hand, applying rules 1 and 2 in an odd plane and rules 4 and 5 in an even plane results in a boe degree of adaptiveness (P_{boe}) as follows:

$$P_{boe} = \begin{cases} \frac{(d_x + q + k)!}{d_x! q! k!} & \text{for even planes} \\ \\ \frac{(h + d_y + k)!}{h! d_y! k!} & \text{for odd planes} \end{cases} \tag{5.13}$$

where q is equal to $\lceil \frac{d_y}{2} \rceil$ or $\lceil \frac{d_y - 1}{2} \rceil$ depending on the row at which y_s lies and d_y.

As opposed to the conventional oe (Eq. (5.12)), which has the x direction constrained in all planes, the balanced oe constrains direction x in an odd plane and direction y in an even plane. This results in different restrictions on the odd xy-plane from those on the even xy-plane. Consequently, the regularity of traffic patterns (and the resulting communication workload) which occur in adjacent layers (due to similar restrictions) is broken as shown in Fig 5.6. This results in a more balanced adaptiveness among the planes which enhances the performance of runtime adaptive selection strategies.

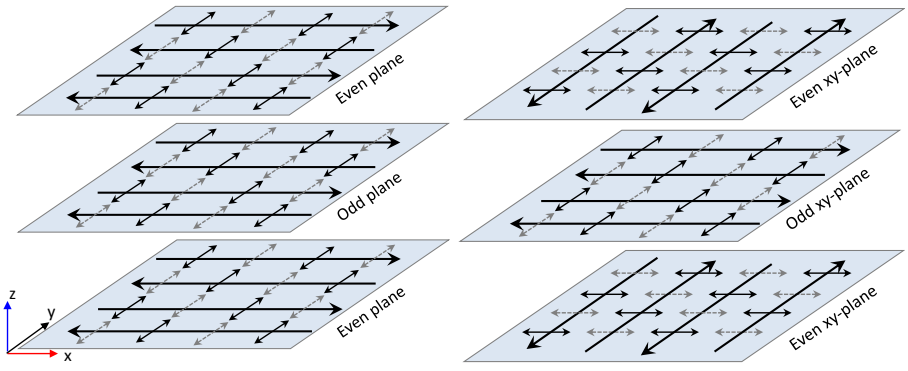


Figure 5.6: Illustration of path diversities for both, (a) conventional 3D odd-even, and (b) the proposed balanced 3D odd-even.

5.4.4 *Coupling DP with 3D-NoC*

The 3D NoC with the DP network coupled, is illustrated in Fig. 5.7. DP network is a network of distributed computational units. The topology of the DP network resembles that of the communication structure of the NoC. At each node there is a computation unit to implement the DP shortest path computation. The solution is propagated to the neighbouring units. The DP network is tightly coupled with the NoC and each computational unit locally exchanges control and system parameters with the corresponding NoC router (as detailed below in Section 5.6.6).

The DP network converges to the optimal solution in a time period which depends on the network structure (the diameter of the network) and the clock frequency of the DP network [128]. This is detailed in the following section. After convergence, the DPN passes the control decisions to the NoC routers. The cost can be communicated across the DP network by dedicated links. This enables fast convergence of the network in response to rapid changes in the cost function. Another option is to use the existing NoC structure to propagate the DP cost. This increases DPN convergence delay. For the present case cost, is defined as the local router temperature, which changes slowly compared to the system clock since the thermal time constant of the chip is usually in the order of milliseconds or seconds while the frequency period is in the order of nanoseconds. This enables the use of the existing NoC structure for DP cost propagation.

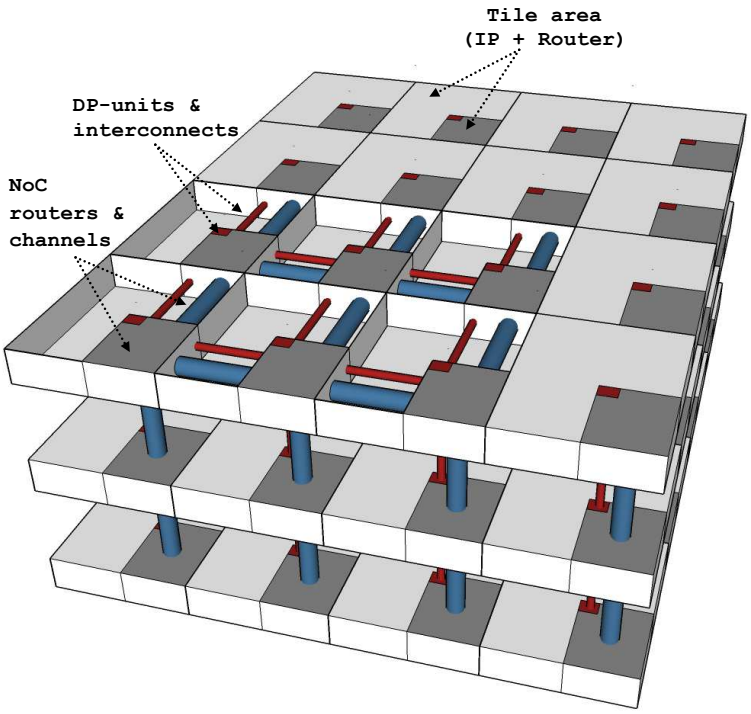


Figure 5.7: A 3D mesh NoC with DPN for coupled.

5.4.5 *DP-Network Convergence Time and Complexity*

The DP-network converges to an optimal routing solution after a delay which is determined by the network topology, the delay of data propagation, and the computational delay of of the DP-unit. Each unit involves $\mathcal{O}(|\mathcal{A}|)$ additions and comparisons where $|\mathcal{A}|$ is the number of edges in the unit. Note that the number of additions corresponds to the number of adjacent nodes. Hence, the worst case solution time is $\mathcal{O}(i|\mathcal{A}|)$, where i is the number of iterations evaluated by each unit.

In software computation, the number of iterations, i, which guarantees that all nodes have been updated, equals the number of nodes in the network, i.e. $i = |\mathcal{V}|$ [49]. However, in hardware implementation with parallel execution, i is determined by the network structure and $\mathcal{A}$ additions can be executed in parallel. Each computational unit can simultaneously compute the new expected cost for all neighbouring nodes. The network convergence time is proportional to the network diameter, which is the longest path in the network. To determine the minimum clock frequency of the DPN that guarantees the convergence, the following condition must be met:

$$N_{dim} \times N \times (t_{link} + t_{unit}) < t_{temp\_sampling}, \tag{5.14}$$

where N_{dim} is the NoC diameter, N is the number of NoC nodes (number of destinations), t_{link} is the delay time of the DP-interconnect, t_{unit} is the delay time of the DP-unit computation, and $t_{temp\_sampling}$ is the time period for temperature sampling. This condition can yield an upper bound of t_{unit} from which the minimum frequency of the unit can be computed.

5.5 DYNAMIC THERMAL MODELLING FOR 3-D NOCS

A traffic and thermal co-simulation tool is developed in this work forthe dynamic thermal modelling of the 3D NoC in order to evaluate the proposed routing strategy. This tool comprises traffic, power, and thermal models. These models are integrated in an automated flow. Fig. 5.8 illustrates the input configuration files and parameters required by the model. This figure also depicts the model components and the computational flows among them. The technology, architecture, and packaging parameters are used to configure the tool. The results, in terms of cycle-accurate temperature variations over a discrete sampling interval, are saved in computer storage. The models used in this traffic and thermal co-simulation tool are described in this section.

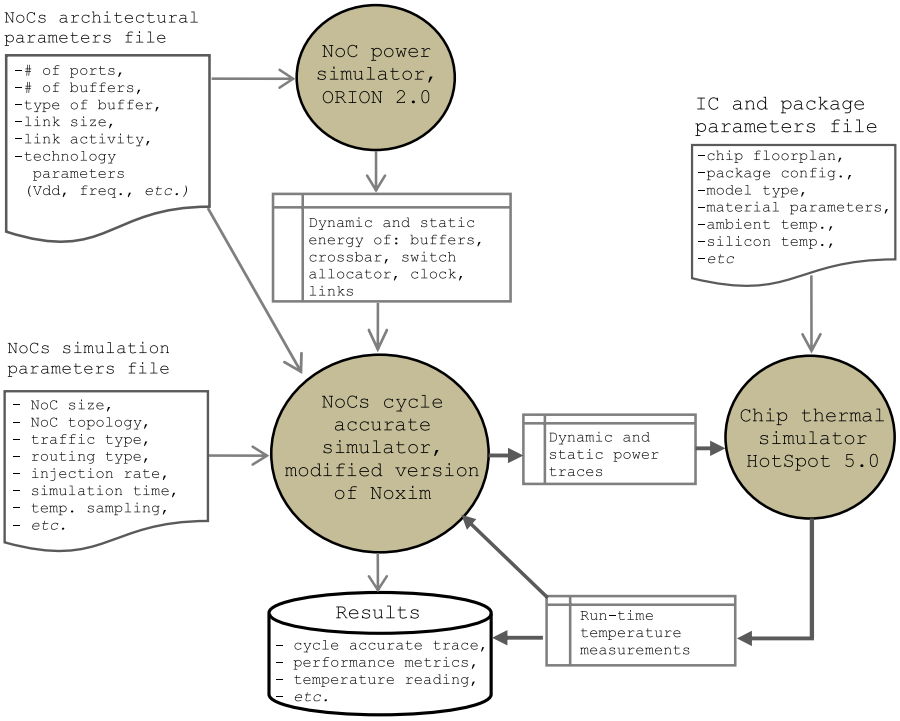


Figure 5.8: Automated computational flow of the proposed tool for dynamic thermal optimization for 3D NoC.

5.5.1 *Traffic Model*

The network of concern in this work is a collection of router nodes connected by channels. Each router is connected to a single IP core that injects and consumes packets. A wormhole flow control technique [62] is used and the configurations of Intel's TeraFLOPS chip [193] is adopted with an input buffer size of 16 flits and a packet size of 3 flits.

Traffic simulation is performed using a modified version of *Noxim* [70]. The router architecture is modified by adding additional ports for communicating the DP costs, and a DP-based routing-table updating algorithm (Algorithm 5.1) is introduced. Moreover, the 2D NoC routing algorithms and traffic patterns are modified to support the proposed 3D NoC routings and traffic patterns. The power model of the original simulator is updated with power for both the router and the DP unit. Router power and area are evaluated using a NoC power simulator, while DP unit power and area are evaluated using a hardware synthesis tool.

5.5.2 *Area and Power Model*

Router power and area are computed using the NoC power and area model *ORION 2.0* [112]. The power traces of the computational units and floorplan are taken from the Intel TeraFLOPS chip [193]. In general, the energy dissipated by the NoC is divided into the following

categories: 1) routing and arbitration; 2) flit forwarding energy; 3) flit receiving energy; 4) clock energy and 5) leakage energy. The flit receiving energy is assumed to be equal to buffer writing energy. The forwarding energy, which dominates the energy consumption of the router, comprises the energies of buffer read, E_{buffer}, crossbar traversal, $E_{crossbar}$, and link traversal, E_{link}:

$$E_{forward} = E_{buffer} + E_{crossbar} + E_{link}. \tag{5.15}$$

The flit forwarding energy along the vertical direction is assumed to be the same as that for the horizontal direction except for the link traversal energy E_{link}. The E_{link} of the vertical link is computed assuming a through-silicon-via (TSV) link length equal to layer thickness.

The energy consumption of any computational unit U (E_U) in the tile is assumed to be modulated by its communication energy ($E_{U\_local\_comm}$). This energy dynamically changes according to local data transfer (data transfer from and to the local router). Thus, the energy of the computational unit E_U is computed as:

$$E_U = \beta . E_{U\_local\_comm} \tag{5.16}$$

where β is the ratio of the communication power to the computation power of unit U. U is any tile unit other than the router. For the TeraFLOPS, these units are: data memory (DMEM), instruction memory (IMEM) and floating point units (FPMAC0 and FPMAC1). β is estimated based on the results of communication and computation powers for Intel's TeraFLOPS CMP [193].

5.5.3 *Thermal Model*

A typical modern chip package consists of several layers. Fig. 5.9 illustrates these layers for a typical ceramic ball grid array (CBGA) package of a 3D IC with four vertically-stacked silicon dies. This is the packaging scheme adopted in this chapter. The package has several heat conduction layers including heat sink, heat spreader, thermal paste, silicon die (s), C4 pads, ceramic packaging substrate, and solder balls [96]. These layers are designed in such a way as to maximize the heat-flow from the active layer(s), or silicon die(s), to the ambient. This path represents the primary heat-flow in the package. Thus, the heat generated by chip activity could be removed efficiently.

The heat-spreader and heat-sink layers are often made of aluminium, copper, or some other materials of high thermal conductivity. In addition to the primary heat flow path to the heat sink, there is a secondary heat-flow path from the die(s) to the package, and the PCB is designed in such a way as to minimize the heat-flow in order to protect the board and other installed devices from excessive heat accumulation.

To model all of these heat transfer paths, the thermal resistance and capacitance (RC) model *HotSpot* [98] is employed. This model is built on top of the dualism between thermal and electrical phenomena, as both are described by the same differential equations. Thermal resistance (R) and thermal capacitance (C) can be computed from the following equations:

$$R = \frac{t}{k \times A} \tag{5.17}$$

$$C = c \times t \times A \tag{5.18}$$

where t is the thickness of material in m, k is the material's thermal conductivity per unit volume (in $W/(m.K)$), A is the cross-sectional area (in m^2) and c is the thermal capacitance per unit volume (in $J/(K.m^3)$).

Die-level thermal RC modelling can be done at the functional unit level or at finer levels, where the die is divided into regular grid cells in order to gain a more detailed temperature distribution [96]. These cells represent the different architectural blocks in the die. Likewise, the chip can have several dies stacked on top of each other in a 3D IC, and this can be readily added to the thermal RC model. A more detailed discussion regarding thermal model derivation and calibration, and its validation against a commercial finite element simulator can be found in the original HotSpot papers [172, 96, 97].

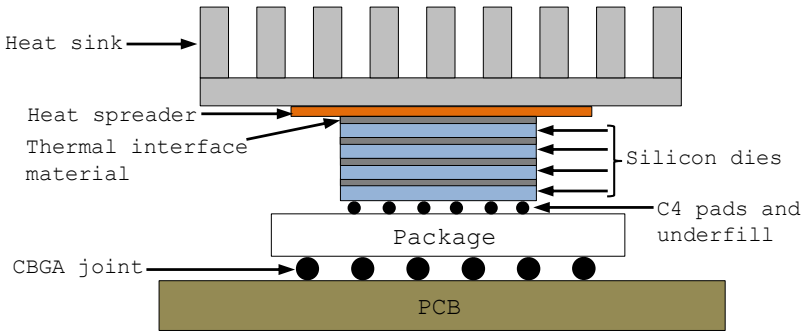


Figure 5.9: Illustration of various layers in a typical ceramic ball grid array (CBGA) package of a 3D IC with four layers [96].

5.6 RESULTS AND DISCUSSION

5.6.1 *Experimental Setup and Tools*

To evaluate the proposed 3D NoC thermal optimization, a 3D NoC-based CMP is considered. Tile area and power are computed using the results presented in a previous work [193]. The power and area

of the router unit is modified to account for the extension to the 3D mesh router, which consists of seven input/output channels and the overhead introduced by the DP-unit.

For thermal model configuration, layer die thickness is set to $0.15mm$. The thermal resistivity of the heat sink is $0.0025\ mK/W$. The dies are separated by an interlayer material with a thickness of $0.02\ mm$. The heat spreader is placed on top of the silicon dies. The thermal interface material (TIM) is used as the filler material in order to separate the heat spreader and the silicon dies. The resistivity of the interlayer material is set to $0.25\ mK/W$. Ambient temperature is assumed to be $25°C$. V_{DD} and frequency are assumed to be 1 V and 3 GHz, respectively. Table A.1 in Appendix A gives details of the various package layers used in this work, including dimensions and material parameters.

For synthetic traffic simulation, the traffic patterns considered are *Uniform*, *Transpose*, and *Hotspot*. For Random traffic, each tile sends data to all other tiles with equal probability. For the Transpose case, $tile(i, j, k)$ sends packets to $tile(X - i, Y - j, Z - k)$, where X, Y and Z are the x, y and z dimensions of the NoC, respectively. For the Hotspot traffic pattern, the four central tiles of the top layer (layer 3) receive an extra 5% in addition to the Uniform (random) traffic. For each of these traffic patterns, the floorplan is arranged as a 3D mesh with a size of $6 \times 6 \times 4$.

In addition to the above traffic scenarios a scenario is included which simulates a layer of shared memory. This is an important application for 3D stacking [33]. To evaluate the proposed RTM in such scenario, the simulator is modified to generate traffic that mimics 3D memory stacking by assuming that top layer is a memory resource shared by the layers of computational cores and which receives 30% of the traffic of the these layers. This traffic is called *Memory-Wall* here.

The following schemes are evaluated:

- Odd-even with buffer selection (oe_buff): The original odd-even routing (Rules 1 and 3 are applied for all planes) with buffer level selection strategy and no thermal optmization.
- XYZ with downward routing (dw_xyz): The RTM scheme proposed in [42] for thermal optimization in 3D NoCs, which uses traffic-aware downward routing.
- Odd-even with DP selection (oe_dp): The original 3D odd-even routing with a DP guided selection strategy for thermal optimization.
- Balanced odd-even with DP selection (boe_dp): Odd-even (Rules 1 and 2) is applied in an even plane, modified odd-even (Rules 4 and 5) is applied in an odd plane, and Rule 3 is used for vertical turns with a DP guided selection strategy for thermal optimization.

For traffic-aware downward routing (dw_xyz), the authors proposed the use of a downward level which depends on the packet injection rate and traffic type and, thus, requires calibration. However, in this study, considerable effort is expended in the calibration of the downward level for different traffic patterns and PIRs to ensure fair comparison. This is done according to the method proposed in their paper to achieve similar performance results [38].

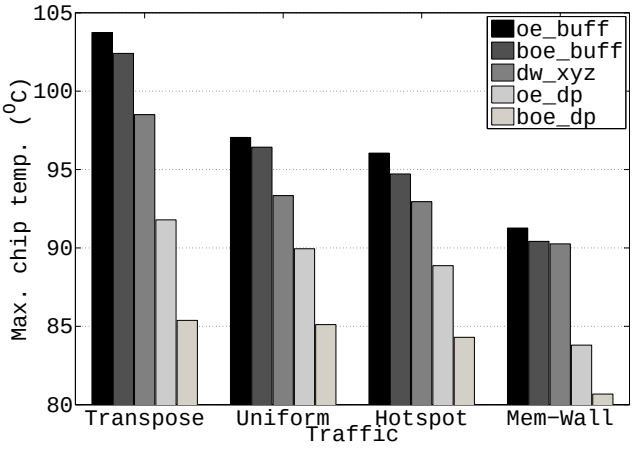
5.6.2 *Temperature Results*

In the first experiment, simulations are run until the chip temperature is stable after 5 million cycles. The temperature results are illustrated in Fig. 5.10 for a PIR of 0.008 packet/cycle/IP. For this PIR, all the routing schemes considered achieve the same throughput of 0.0481 flits/cycle/IP. This guarantees a fair comparison in terms of the resulting temperature.

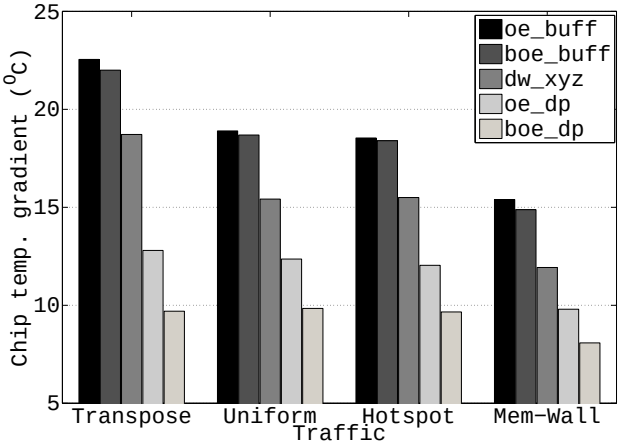
The maximum chip temperature and the spatial temperature gradient, which is the difference between the maximum and the minimum temperatures, are shown for the four schemes considered . The results for balanced odd-even with buffer selection (boe_buff) are also included here for reference. Taking oe_buffer as a baseline, and given that the four methods have similar throughput, it can be noticed that oe_dp outperforms dw_xyz in its thermal behaviour for all the considered traffic patterns. However, boe_dp outperforms both dw_xyz and oe_dp for the four traffic scenarios. For instance, in this experiment, boe_dp achieves more than 18°C cooling compared to the oe_buffer, while dw_xyz could only achieve 5°C cooling for the Transpose traffic case (Fig. 5.10a). Moreover, it can be noticed that the spatial temperature gradient in the chip for boe_dp is nearly half of that of dw_xyz for all the traffic patterns considered . This implies that higher thermal balancing is achieved by the proposed scheme (see Fig. 5.10b).

Figures 5.11 and 5.12 illustrate the spatial temperature and NoC power distributions, respectively, for the four schemes under the Transpose traffic case. Fig. 5.11 indicates that, besides its higher cooling performance boe_dp, achieves a more homogeneous spatial thermal distribution compared to dw_xyz. The thermal behaviour results can be explained by the power distribution results in Fig. 5.12. The cooling performance of an RTM technique is determined by its capability to accommodate to the cooling system of the chip. In this scenario it can be seen that the proposed boe_dp can migrate power consumption towards the heat sink more efficiently compared to dw_xyz (see Figures 5.12d and Fig. 5.12c). As a result, better thermal moderation can be achieved by the proposed approach.

To see how each of the considered schemes behaves when yhe PIR changes, Fig. 5.13 shows the maximum and gradient of temperature for a PIR range of 0.004-0.016 packet/cycle/node with Transpose traf-



(a) Max. temperature



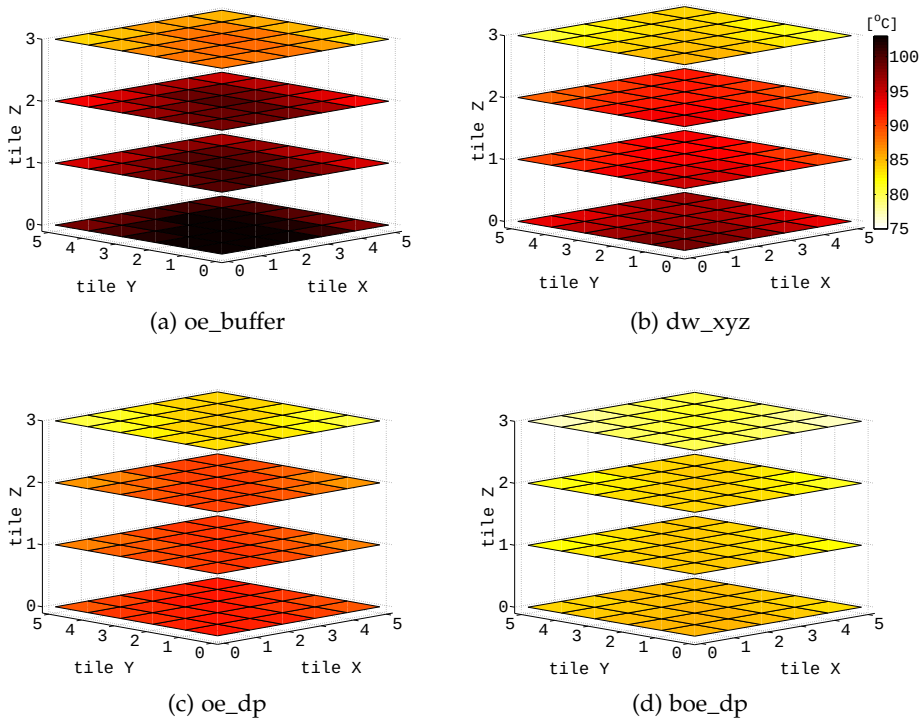
(b) Temperature gradient

Figure 5.10: Comparision of the maximum and spatial gradient (min.-max.) of chip temperature for the considered routing strategies with various traffic scenarios.

fic. In this range of PIR, all of the routing schemes achieve the same throughput. As expected, both the maximum and gradient of temperature significantly increase with PIR for all schemes. However, the trend found in Fig. 5.10 can also be seen here. Both DP schemes (boe_dp and oe_dp) outperform dw_xyz in terms of both the maximum and gradient of chip temperature for all the considered PIRs.

These results can be explained by the fact that dw_xyz adapts only to inter-layer thermal variations and not intra-layer thermal variations. Thermal variations within the same layer can be significant, but they are not exploited by dw_xyz. Moreover, dw_xyz is tailored to a cooling system in which the heat sink is placed above the chip. On the other hand, our scheme achieves better results because DP can adapt to inter-layer as well as intra-layer thermal variations due to its adaptive distributed nature.

Moreover, Figures 5.10, 5.11 and 5.13 also indicate that boe_dp considerably outperforms oe_dp in terms of both the maximum and gradient of the temperature. This clearly indicates that the higher balancing of adaptiveness offered by the balanced 3D odd-even routing (boe_dp), compared to the conventional 3D odd-even (oe_dp), is effective in improving the efficiency of the DPN in heat diffusion.



(a) oe_buffer (b) dw_xyz

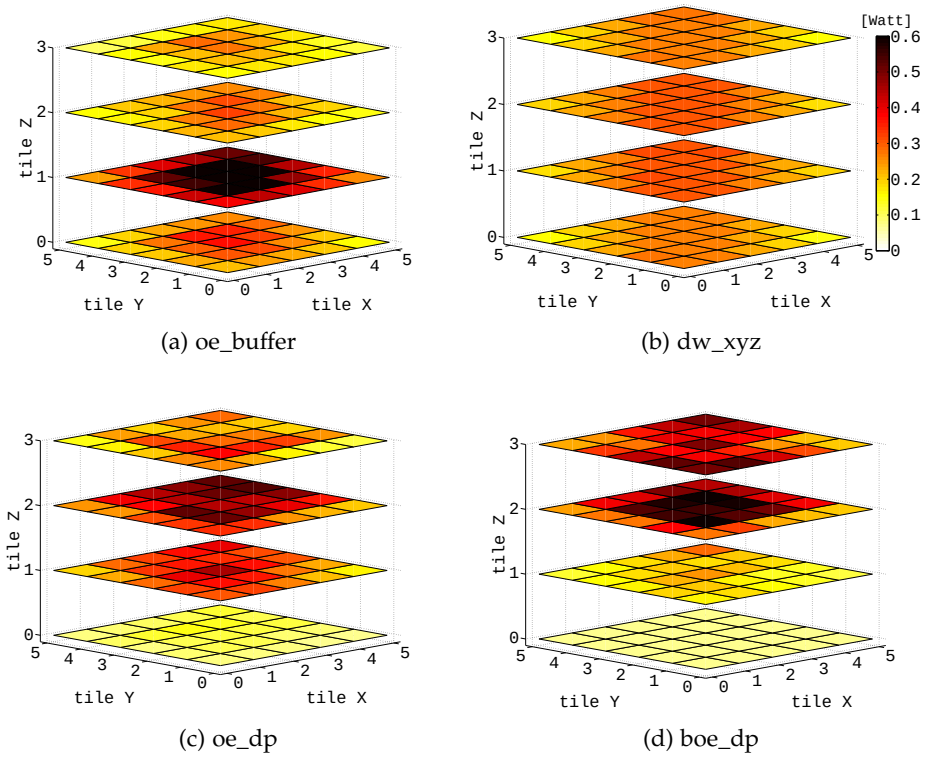
(c) oe_dp (d) boe_dp

Figure 5.11: Spatial thermal distributions (°C) for the four routing strategies.

5.6.3 *Reliability Improvement*

To give an insight into the implications of the thermal optimization schemes considered on reliability improvement, the FIT values for different fault mechanisms are evaluated. FIT_{EM}, FIT_{SM}, FIT_{TDDB} and FIT_{TC} values are computed using equations (5.1), (5.2), (5.3) and (5.5), respectively. The total FIT (FIT_{TOT}) is computed using Eq. (5.7). The material-dependent parameters in these equations are taken from a previous work [174], while the constants (A_{EM}, A_{TDDB}, A_{SM} and A_{TC}) are taken from another work [175] assuming 65nm technology. The temperature results presented in Fig. 5.10 for PIR $=$ 0.008 *packet/cycle/node*, are used in FIT computation. The FIT results for the four thermal optimization methods with the four traffic patterns are shown in Table 5.1.

It can be seen that, in general, any reduction in chip temperature results in lower FIT, better reliability and longer chip lifetime. Taking



(a) oe_buffer (b) dw_xyz

(c) oe_dp (d) boe_dp

Figure 5.12: Spatial power distributions (W) for the four routing strategies.

the oe_buff as a baseline, it can be seen that the higher temperature reduction achieved by boe_dp compared to the other methods leads to a significant improvement in reliability. For instance, boe_dp achieved 18.4°C temperature reduction compared to only 5.2°C for dw_xyz in the Transpose traffic case. This translates to a 63.46% reduction in FIT_{TOT} for boe_dp compared to only 25% for dw_xyz. This indicates the crucial significance of the thermal optimization achieved by the proposed scheme in increasing reliability and IC lifetime.

5.6.4 *Performance Results*

Fig. 5.14 compares the performance of the four schemes in terms of average network delay versus the achievable throughput curves under the four traffic scenarios. It can be noticed that, in general, the performance of dw_xyz is considerably lower than that of the other schemes. Also, it can be seen that the performance of oe_dp and boe_dp is nearly the same, and both slightly outperform the oe_buff for the Uniform and Transpose traffic cases, while the oe_buff is better than the thermal-aware DP approaches for the Hotspot and Memory Wall traffic patterns. However, for the latter two traffic patterns, both DP approaches are still better than dw_xyz.

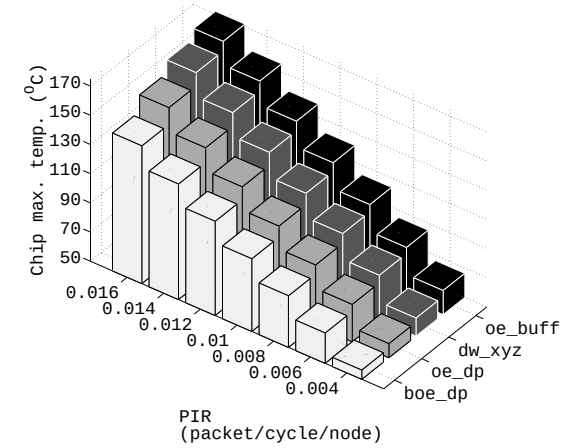
Traffic	Routing strategy	FIT_{EM}	FIT_{SM}	FIT_{TDDB}	FIT_{TC}	FIT_{tot}	Improvement
Transpose	oe_buff	33,268	7,360	21,158	3,897	65,684	-
	dw_xyz	22,509	5,526	17,770	3,315	49,120	25.2%
	oe_dp	13,426	3,744	14,155	2,647	33,972	48.3%
	boe_dp	8,049	2,521	11,342	2,088	24,000	63.5%
Uniform	oe_buff	20,162	5,091	16,924	3,163	45,341	-
	dw_xyz	15,154	4,106	14,924	2,794	36,977	18.5%
	oe_dp	11,614	3,350	13,288	2,479	30,731	32.2%
	boe_dp	7,874	2,478	11,236	2,066	23,654	47.8%
Hotspot	oe_buff	18,679	4,808	16,362	3,061	42,911	-
	dw_xyz	14,700	4,012	14,727	2,756	36,196	15.7%
	oe_dp	10,659	3,136	12,803	2,383	28,980	32.5%
	boe_dp	7,371	2,352	10,922	2,002	22,647	47.2%
Mem. W.	oe_buff	12,889	3,629	13,905	2,599	33,021	-
	dw_xyz	10,283	3,050	12,605	2,344	28,281	14.4%
	oe_dp	7,075	2,278	10,733	1,962	22,048	33.2%
	boe_dp	5,466	1,857	9,616	1,726	18,866	42.8%

Table 5.1: Comparison of FIT due to different fault mechanisms for the four routing strategies and different traffic patterns for a $6 \times 6 \times 4$ 3D NoC configuration.

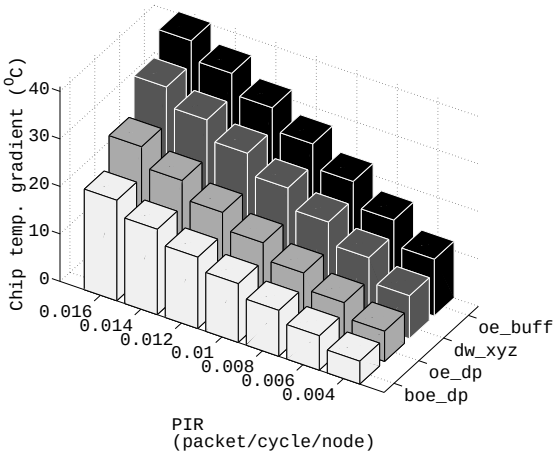
Another experiment is conducted to evaluate the performance of the thermal optimization methods considered. The network throughput and the average delay that causes the first violation of a thermal limit are recorded. Table 5.2 summarizes the results of this experiment for the traffic patterns considered and with three thermal limits: 60°C, 70°C and 80°C. The thermal limits are chosen in the reasonable working temperature range and in line with previous work [170]. This table also shows the boe_dp throughput improvement over oe_buff and dw_xyz. For instance, given a thermal limit of 80°C, the boe_dp routing can achieve 22% and 41% higher throughput compared to dw_xyz and oe_buff, respectively, for Transpose traffic. This clearly demonstrates the capability of DP to manoeuvre packets dynamically at runtime and to exploit the coolest paths. As a result, the thermal violation is delayed to higher PIR and throughput. Similar trend can be seen for other traffic patterns and thermal limits.

5.6.5 *Real Application Benchmarks*

In this section, the proposed DP-based RTM is evaluated with real application benchmarks. Six real benchmarks with different sizes, topologies and bandwidth requirements are used. These benchmarks include a generic complex MultiMedia system which comprises an h263 video encoder and an mp3 audio decoder (MMS) [92], a telecommunication benchmark (TELE) and a video object plane decoder (VOPD) [137]. In



(a) Max. temperature

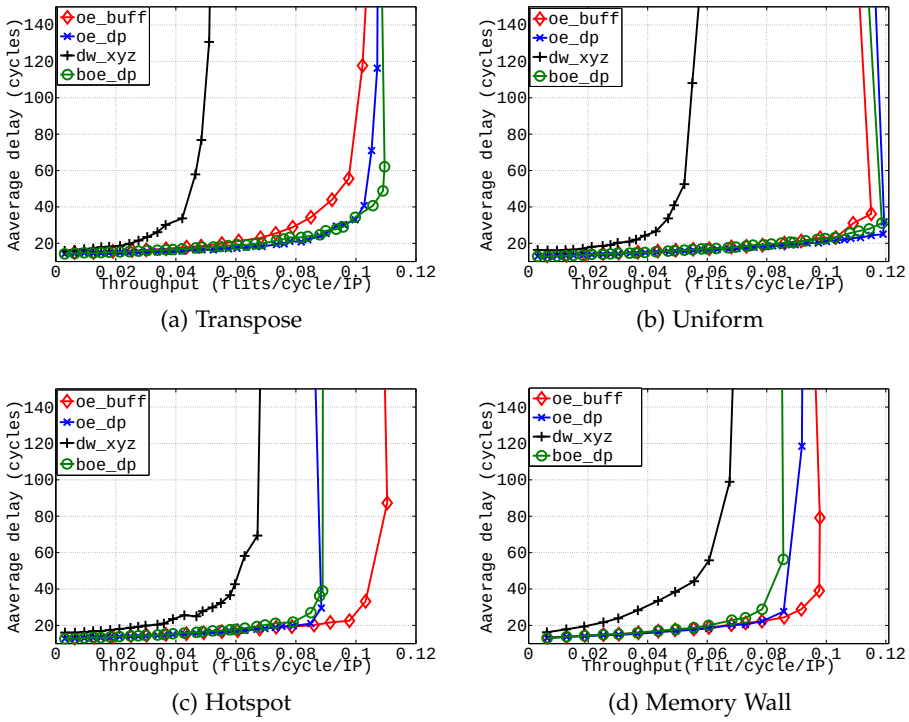


(b) Temperature gradient

Figure 5.13: Maximum and gradient (min.-max.) of chip temperature variation with PIR for the four routing strategies with Transpose traffic.

addition are three benchmarks, AMI49, AMI25 and MPEG4, which were extracted from the Microelectronics Centre of North Carolina benchmark suite found in [7]. Details of the size and communication bandwidth requirements of these benchmarks are shown in the first three columns of Table 5.3.

Mapping these applications to NoCs is achieved using the algorithm proposed a previous work [53]. However, the small sizes of these benchmarks mean that they are not suitable to be used in a 3D NoC platform of the appropriate size. Therefore, the sizes of these benchmarks are extended to four layers by mapping each task to a pillar (four IP cores aligned on top of each other) rather than a single IP core. The inter-layer traffic is generated by dividing the original bandwidth between communication pairs to the replicated vertical IP cores in the resulting pillar pairs of the 3D NoC platform.



(a) Transpose (b) Uniform

(c) Hotspot (d) Memory Wall

Figure 5.14: Performance comparison of the considered routing strategies in terms of delay versus throughput curves for different traffic scenarios.

The proposed DP-based RTM (boe_dp) is evaluated by comparing it to a previous work [38] which denoted by dw_xyz. The two routings are compared in terms of maximum, T_{max}, and gradient, $T_{gradient}$, of chip temperature, as well as energy consumption, after draining 5MB of data for each benchmark. These results, in addition to the percentage reductions in maximum temperature, temperature gradient and energy consumption achieved by the proposed boe_dp over the dw_xyz, are shown in Table 5.3.

The results in Table 5.3 indicate a significant improvement with the boe_dp over the dw_xyz in terms of temperature regulation, where it could achieve up to 13% reduction in maximum chip temperature and up to 27% reduction in temperature gradient. Moreover, it can be seen that, besides better thermal regulation, boe_dp consumes less energy (up to 6.8%) for the same drained data volume compared to dw_xyz even after adding the energy consumption of the DP units. This is due to the fact that dw_xyz is a non-minimal routing and, thus, consumes higher energy compared to boe_dp. Meanwhile, dw_xyz adapts only to vertical paths by employing non-minimal routing to avoid hot layers.

Table 5.2: Achievable performance metrics with different routing algorithms for different thermal limits and traffic patterns.

Traffic	Routing strategy	Thermal limit °C					
		60		70		80	
		delay (cycle)	thrpt.(flit/ cycle/IP)	delay (cycle)	thrpt.(flit/ cycle/IP)	delay (cycle)	thrpt.(flit/ cycle/IP)
Transpose	oe_buff	17.8	0.0361	18.4	0.0466	21.7	0.0583
	dw_xyz	57.8	0.046	820.9	0.0554	3887	0.0671
	oe_dp	16.4	0.046	17.0	0.0574	19.3	0.0735
	boe_dp	17.9	0.052	19.7	0.0665	23.2	0.0817
boe_dp throughput improvement	vs. dw_xyz	13%		21%		22%	
	vs. oe_buff	44%		43%		41%	
Uniform	oe_buff	15.1	0.0367	16.18	0.0496	17.9	0.0663
	dw_xyz	40.8	0.0488	356.9	0.0630	553.6	0.0643
	oe_dp	15.2	0.0470	16.4	0.0602	17.8	0.0756
	boe_dp	16.2	0.0523	17.5	0.0670	19.6	0.0815
boe_dp throughput improvement	vs. dw_xyz	8%		7%		27%	
	vs. oe_buff	43%		36%		23%	

to be continued...

Table5.2 (continued): Achievable performance metrics with different routing algorithms for different thermal limits and traffic patterns.

Traffic	Routing strategy	Thermal limit °C					
		60		70		80	
		delay (cycle)	thrpt.(flit/ cycle/IP)	delay (cycle)	thrpt.(flit/ cycle/IP)	delay (cycle)	thrpt.(flit/ cycle/IP)
Hotspot	oe_buff	15.1	0.0364	16.5	0.0536	18.0	0.0689
	dw_xyz	27.7	0.0489	58.1	0.0629	182.8	0.0683
	oe_dp	15.82	0.0488	17.2	0.0629	19.6	0.0790
	boe_dp	17.0	0.0522	20.2	0.0697	27.0	0.0850
boe_dp throughput improvement	vs. dw_xyz	7%		11%		25%	
	vs. oe_buff	44%		31%		24%	
Mem. W.	oe_buff	15.80	0.0367	17.33	0.0553	19.04	0.0681
	dw_xyz	38.44	0.0492	98.92	0.0675	554.16	0.0751
	oe_dp	16.96	0.0495	20.20	0.0683	22.33	0.0785
	boe_dp	18.73	0.0555	24.18	0.0728	56.29	0.0856
boe_dp throughput improvement	vs. oe_dp	13%		8%		14%	
	vs. dw_xyz	51%		40%		26%	

Table 5.3: Real benchmarks results. Chip maximum, T_{max}, and gradient, $T_{gradient}$, of temperature, NoC energy consumption for dw_xyz [38] and boe_dp in addition to the percentage improvement of boe_dp after draining 5MB of data.

bench. (size)	NoC size	bw MB/s	dw_xyz			boe_dp			boe_dp Imprv.		
			T_{max} (°C)	$T_{gradient}$ (°C)	E (mJ)	T_{max} (°C)	$T_{gradient}$ (°C)	E (mJ)	T_{max} (%)	$T_{gradient}$ (%)	E (%)
AMI49	$7 \times 7 \times 4$	5,061	116.4	31.5	8.2	104.4	22.7	7.7	13.0	27.7	6.5
AMI25	$5 \times 5 \times 4$	3,680	91.0	27.9	6.4	84.1	23.6	6.2	10.4	15.5	6.7
MMS	$5 \times 5 \times 4$	628	69.3	16.8	6.3	65.7	14.9	6.0	8.1	11.3	5.0
TELE	$4 \times 4 \times 4$	998	79.2	20.1	6.3	72.9	16.3	5.9	11.5	19.0	6.8
VOPD	$4 \times 4 \times 4$	3,731	93.3	25.8	7.8	88.3	22.6	7.5	7.4	12.2	4.0
MPEG4	$3 \times 3 \times 4$	1,951	80.2	20.8	5.7	73.9	16.8	5.5	11.4	19.4	3.7
							Average		10.3	17.5	5.5

On the other hand, the proposed boe_dp adapts to planar paths as well as vertical ones. Paths on the same layer can exhibit significant thermal gradients that are exploited by the proposed technique, due to its global awareness of temperature and distributed control nature, in order to achieve better thermal regulation while adhering to minimal path routing.

5.6.6 *Hardware Implementation*

The DP unit can implemented using different methods. This section investigates the hardware realization of the DP using synchronous circuits. The aim is to realize a DP hardware that augments the NoC's routers so as to provide an adaptive strategy to diffuse heat throughout the chip geometry. Moreover, the resultant power and area overheads of the DP unit are evaluated.

Fig. 5.15 shows the architecture of a router which enables adaptive thermal-aware routing. The architecture supports 3D mesh NoCs. The router circuit is a state-of-the-art design [151] for a 2D mesh with two additional channels for upper and lower layers (labelled as *Up* and *Down*). The design is augmented by an additional block (depicted by a dotted line) which implements the proposed adaptive strategy. The temperature sensor circuit provides the DP computational unit with the local cost.

The local cost and the costs coming from upstream routers are used to compute the *cost-to-go* which is propagated to all downstream routers. The design consists of combinational circuits. However, the control unit block shown in the figure is a mixture of sequential and combinational circuits. It can be realized using a synchronous counter and a few logic gates. The counter scans the destinations and supplies an address reference the the routing table inside the router such that the DP-unit can successively update all destination decisions in the routing table.

The path cost computation is implemented using the DP units, as shown in Fig. 5.16. These DP-units are connected as a DP-network. The coolest path computation necessitates a minimum operation to evaluate and compare the costs coming from upstream routers. This can be realized using comparators and data multiplexers (Fig. 5.16a). Also, an adder is required to add the local router cost to the optimal cost computed using this circuit. Moreover, another data multiplexer is needed to output the associated action for the minimum expected cost. Therefore, the basic circuit of a DP computational unit for the 3D mesh NoC comprises of one adder, six comparators and six data multiplexers.

Direction costs that are involved in the optimal (minimum) cost computation are filtered by *direction selection* control circuit which is shown in Fig. 5.16b. The six enable signals (corresponding to six

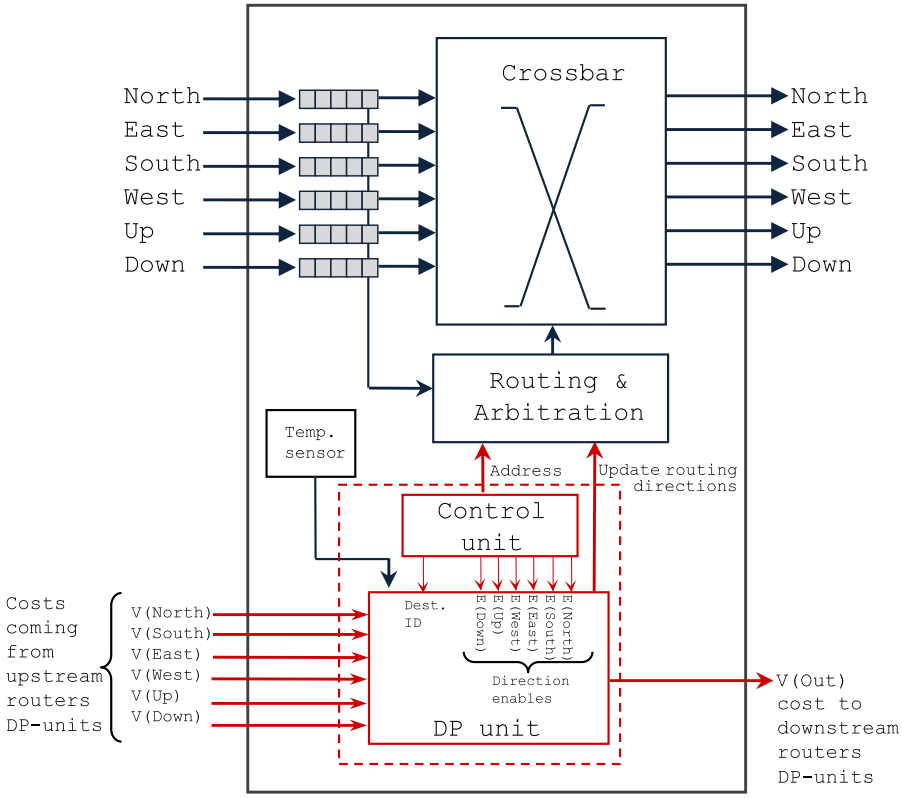
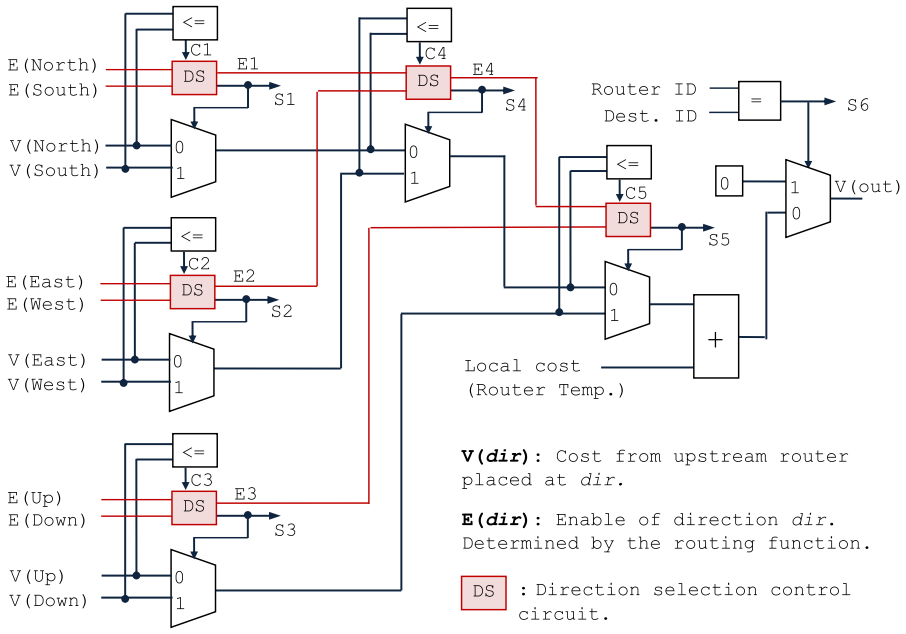


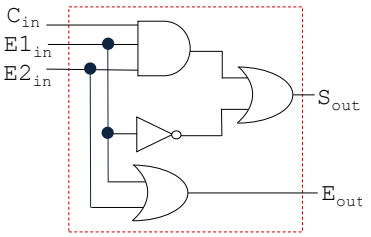
Figure 5.15: Architecture of the 3D NoC router including the DP unit to enable dynamic thermal-aware routing.

directions) used by these control circuits come from the control unit and they are determined by the routing algorithm. The direction selection control circuit discards the direction that is not enabled (where the corresponding enable signal is 0). At each comparison stage, it takes two enable signals (one for each input direction), E_{in1} and E_{in2}, in addition to the comparator result , C_{in}, as inputs. It outputs enable, E_{out}, and selection, S_{out}, signals. S_{out} is equal to C_{out} only if both directions are enabled ($E_{in1} = 1$ and $E_{in2} = 1$). If only one direction is enabled, S_{out} must select this direction regardless of its cost. The enable signal E_{out} equals 1 if any of the input directions is enabled. If neither are enabled, E_{out} is 0 and the cost computed at this comparison stage is discarded at the next comparison stage. The direction selection circuits filter directions and, as a result, only allowable directions (allowed by the routing function) are included in the optimal cost computation. This enables the DP-unit to work in compliance to deadlock-free routing algorithm.

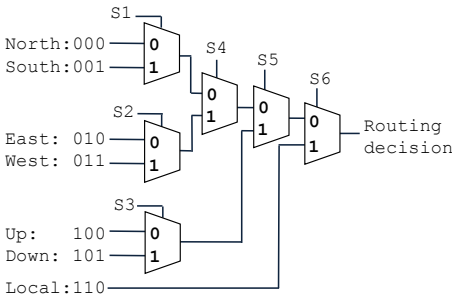
The DP-unit circuit also requires the computation of the optimal routing direction for a particular destination (designated *destination ID*). This direction is used to update the routing table of the *Routing and Arbitration* block. The optimal routing direction computation circuit is illustrated in Fig. 5.16c. This circuit takes the direction selection signals (S_1 - S_6) that result from the optimal cost computation circuit,



(a) Optimal cost computation circuit.



(b) Direction selection control circuit.



(c) Optimal direction computation circuit.

Figure 5.16: Hardware realization of the DP unit.

as multiplexer selects. It requires six 3-bit data multiplexers, assuming seven input directions (six direction plus the local).

Evaluating the hardware power and area overheads for any new proposed solution involved in NoC design is essential. This gives an insight into the trade-offs that exist between the costs paid in terms of power and area and the benefits gained from the proposed technique. To evaluate the area and power overheads of the proposed DP-based routing, the DP computational unit is implemented in Verilog. The implementation is then synthesized using the Synopsys Design Compiler and mapped onto the Faraday UMC 65nm technology library. Table 5.4 summarizes the results for the area and power estimations of the router with the DP-unit for 2-D, 6×6, and 3-D ,$6 \times 6 \times 4$, NoC meshes. The area and power overheads of the DP-unit, as percentages of router total area and power, respectively, are also shown in Table 5.4. The total router area and power have been evaluated using ORION 2.0 [112].

NoC size	Router only		Router+DP unit		DP unit overhead	
	Area (mm^2)	Power (w)	Area (mm^2)	Power (w)	Area	Power
6×6 (2D)	0.4751	0.2168	0.4774	0.2171	0.395%	0.102%
$6 \times 6 \times 4$ (3D)	0.7520	0.3290	0.7644	0.3305	1.645%	0.456%

Table 5.4: Synthesis results: Router and DP-unit power and area in addition to DP unit relative overhead.

It can be noticed that the overhead for both area and power slightly increases with the increase in the NoC dimensions due to the higher table size needed by table-based routing. However, this overhead is insignificant compared to the total area and power of the router. For instance, the area overhead is 1.64% and the power overhead is 0.45% for the 3-D NoC with a size of 144 tiles ($6 \times 6 \times 4$).

It is worth mentioning here that although the router frequency is assumed to be 3GHz , the a DP-unit does not need to operate at this frequency. As described in Section 5.4.5, DP-unit's minimum frequency that guarantees convergence can be estimated from NoC size and the temperature sampling period. Using Eq. 5.7 and assuming an NoC size of $6 \times 6 \times 4 = 144$ and an NoC diameter of 16 with a 10μs temperature sampling time, clocking the DP-unit with 200MHz guarantees the convergence of all DP units within the sampling period of the temperature.

5.7 SUMMARY AND CONCLUSION

Due to aggressive technology scaling and migration to multi-layer 3D VLSI, future 3D NoC-based systems will face serious challenges, the most significant of which is the thermal challenge. In 3D NoC-based CMPs, communication contributes significantly in heat generation and could modulate the whole chip activity. In the present work, an adaptive distributed thermal optimization strategy for 3D NoCs is proposed. This uses distributed DP units connected via a dynamic programming network to manage the routing workload at runtime to achieve global thermal moderation. As a result, the routing adapts such that the heat is diffused from the 3D chip geometry and thermal hotspots are minimized. The dynamic programming network is improved such that the cost computation and propagation takes place in compliance with the deadlock-free routing algorithm. Furthermore, 3D adaptive routing algorithms are modified to improve path diversity, balance adaptiveness and increase DPN performance. The proposed method has been rigorously evaluated and the results show that it outperforms recently

proposed RTM methods in terms of adaptation efficiency, thermal regulation, lifetime reliability and performance. Hardware implementation details are also presented and the overheads introduced by the proposed method are evaluated and reported. These overheads are shown to be insignificant relative to those of the router hardware. This work tackles a major problem in future 3D systems-on-chip and proposes an efficient solution which could provide better thermal integrity in future many-core systems.

6

FPGA IMPLEMENTATION OF THERMAL-ADAPTIVE ROUTING IN NOCS

6.1 INTRODUCTION AND MOTIVATION

The continuous shrinking of technology node is increasing the number of cores that can be placed in a single chip and, in the foreseeable future, systems with thousands of cores in a single chip will be feasible [103]. However, this massive scale of integration comes with many challenges, of which the thermal challenge is the most altering one. There are many reasons for this, such as increased device density, decreased transistors junction temperature and exacerbated spatial temperature gradients. These effects lead worst-case cooling system design to be infeasible [172].

Higher temperatures cause increased leakage and delay in both logic gates and wires due to higher resistivity. Also, higher temperature gradients reduces chip reliability and shortens device lifetime [170].

On the other hand, Networks-on-Chip (NoCs) are proposed as a promising communication paradigm for SoC and CMP, which could overcome the limitations associated with the on-chip bus. Providing scalability, flexibility and power efficiency, the NoC enables the integration of many cores in a single chip [23]. In NoC-based systems, the communication power budget takes up a significant portion of overall chip power and thus could contribute significantly to chip heating in many core systems in the future [33, 169]. Even in recent designs, the NoC has been shown to either exceed, or have comparable contribution to, processors in heat generation, particularly for communication-centric applications. A characterization of the thermal profile in the MIT Raw chip has revealed that the NoC can have a higher thermal than processors in highly parallelizable and communication-centric applications [170]. Another example is Intel's TeraFLOPS 80-core processor, where the power density of the NoC routers is nearly double that of other units in the tile such as floating-point and memory units [193] resulting in a higher contribution to chip heating. Another important point is that controlling network communication traffic offers a unique opportunity to control a workload spanning the majority of the chip area. Thus, better control of routing paths would achieve thermal moderation and control over the entire chip.

Motivated by the above-mentioned factors, various studies have proposed techniques for thermal-aware routing in NoCs [170, 169, 42, 38, 157]. While such techniques are shown to achieve good thermal moderation, these works have many limitations, such as working

offline or not being able to adapt to runtime dynamics [157], as well as adapting only to a particular cooling system [38]. However, the most important limitation of much of this research is that neither details of hardware implementation for the proposed schemes, nor evaluation of the overhead that they introduce are included.

A runtime thermal-adaptive routing strategy for NoCs is presented in Chapter 5. This strategy uses a distributed DP-based architecture to achieve global thermal optimization at runtime for the chip. However, the Chapter 5 focuses on describing the strategy and studying its efficiency by comparing it with other strategies for thermal optimization in NoCs. This chapter, on the other hand, describes the details of implementation challenges associated with the proposed strategy by evaluating it with FPGA hardware implementation. Results show that the proposed hardware implementation of the scheme is highly flexible in manoeuvring the packets away from hot regions, and it can exploit cool paths to adapt effectively to the heat dynamics in the chip with insignificant hardware and performance overheads. The major contributions of this chapter can be summarized as follows:

1. Efficient implementation of the DPN-based runtime thermal-adaptive routing strategy for NoCs proposed in Chapter 5 is presented. Consideration is given to convergence time and ring oscillator (RO)s are used for thermal sensing.

2. Various challenges related to sensor accuracy and precision, such as isolating the IR drop and hardware implementation of sensor models to compensate for the intra-chip process variations, are addressed.

3. The proposed strategy is implemented in FPGA and a rigorous evaluation is performed to determine the effectiveness of the proposed RTM in terms of functionality, thermal regulation and throughput performance.

6.2 RELATED WORK

Thermal modelling and control in VLSI systems has gained a lot of attention in recent years [170, 172, 65, 122, 38]. Skadron and his research group worked on modelling the thermal behaviour of the VLSI chip. Their work relies on the basic thermal RC dynamic compact model for modelling the main heat transfer paths for given package settings [172, 96]. Other researchers focused on thermal control methods. For example dynamic voltage and frequency scaling (DVFS) has been used in to avoid exceeding the emergency temperature [65]. In another work, on-line task allocation for multi-core systems was employed to avoid exceeding the thermal limit as well as hotspot formation [126].

Runtime thermal management schemes for on-chip networks with a reactive strategy that uses traffic throttling to respond to thermal

emergencies have also been proposed [170, 122]. However, exploiting NoC routing to better control chip heat distribution has received only limited attention in existing literature. A routing-based NoC RTM strategy is proposed motivated by the results of the characterization of the thermal profile in the MIT Raw chip which reveals that NoC can surpass processors in heat generation. In the proactive phase of this strategy, neighbouring nodes exchange traffic counters as a means of thermal balancing. When the chip's thermal limit is violated, throttling is proposed as a reactive strategy [170].

Traffic balancing would not necessarily lead to thermal balancing. This is mainly because heat has a relatively high time constant and is regional in nature. Also, temperature changes may arise from sources outside the network itself. Thus, a routing strategy that uses direct temperature readings at runtime is necessary. Moreover, global knowledge of these readings is crucial for effective thermal-aware routing, particularly, when high numbers of cores are integrated in one chip.

Another work [157] has proposed thermal-aware application-specific routing path allocation in 2D mesh MPSoCs. Routing paths are allocated at design time such that thermal variations in the cores are compensated for and thermal hotspots are minimized. The shortcoming of the scheme is that it is an offline technique and cannot adapt to thermal dynamics at runtime, which may be significant.

On the other hand, many works have been published on thermal modelling, characterization and sensing in FPGAs [180, 210, 72, 20, 195]. In one experimental analysis of RO-based thermal sensors in FPGAs is presented [72], the authors showed that a small change in voltage causes a high variation in RO frequency. This implies that voltage variations need to be isolated in order to achieve accurate thermal readings from the sensor. Thermal sensing with ROs requires frequency counters to capture the impact of temperature on RO frequency. A compact implementation of these counters, using a linear feedback shift register (LFSR) counter to reduce the FPGA resources required has been presented [210]. Another work explores the use of the metastability phenomenon in flip-flops to implement thermal sensors that consume less device resources and dos not require high clock frequency [180].

A system for thermal sensing in FPGA-based SoCs has been presented [195]. This system is composed of an array of thermal sensors and a controller which determines when the sensors have to be enabled. It enables the sensors and decides to activate thermal management techniques if needed.

In this work, an NoC dynamic thermal management system design and implementation are presented. This includes both thermal-based control and on-chip thermal sensing design and FPGA implementation. To the best of the authors knowledge, no such detailed design and hardware implementation of a thermal-adaptive routing strategy with

implementation details for both RTM control and thermal sensing has been proposed in open literature.

6.3 BACKGROUND

This section presents a brief description of RTM schemes, in addition to introducing two important concepts related to RTM; namely, RTM control techniques, and on-chip thermal sensing techniques.

6.3.1 *Thermal Optimization and Management*

Severe thermal challenges has arisen due to the continuous shrinking of the feature size. Thus, in design-time thermal optimization it is becoming increasingly difficult to guarantee the thermal integrity of VLSI systems [170]. As a result, RTM techniques are becoming necessary. These techniques would diffuse heat and regulate the system's operating temperature at runtime to ensure a safe thermal range such that chip and package design for typical cases would be possible.

RTM techniques monitor the temperature at runtime and alter system behaviour accordingly. These techniques can be divided into two categories: reactive and proactive. Reactive techniques work when the thermal limit is exceeded and sacrifice performance in order to achieve thermal regulation, for example DVFS. On the other hand, proactive techniques try to reduce thermal hotspots and minimize the temperature at runtime. This reduces, and may alleviate, the need for reactive action, thus improving both chip performance and reliability. Examples of these techniques are dynamic task scheduling and allocation in the CMP [86, 91, 50, 126].

6.3.2 *On-Chip Thermal Sensing*

Dynamic runtime thermal control and management applications require high performance on-chip thermal sensing. Various techniques can be used for on-chip temperature sensing. Analogue sensors are accurate and easy to calibrate. However, they require A/D conversion and analogue implementation, which involve high overheads and cost associated with using both analogue and digital technologies in the same chip.

Alternatively, in digital systems, thermal sensing can be achieved by measuring the impact of temperature on the delay of CMOS gates. Higher temperatures increase the propagation delay of CMOS gates and this delay can be characterized by the oscillation frequency of an RO.

An RO is a loop of an odd number of inverters. The oscillation frequency of an RO (f_{RO}) is given by the following formula:

$$f_{RO} = \frac{1}{2 \times n \times t_d} \tag{6.1}$$

where n is the number of inverters and t_d is the delay of a single inverter. Since t_d is directly proportional to temperature, f_{RO} is inversely proportional to temperature [210]. However, t_d varies depending on other physical parameters, mainly V_{DD} fluctuations and process variations. Thus, before relying on RO counts to measure temperature, these parameters must be isolated.

6.4 METHODOLOGY

In this section, the design trade-offs and implementation details of the proposed RTM are presented. The proposed RTM is a proactive technique for NoC-based CMPs. It achieves thermal minimization by employing a runtime dynamic routing strategy which adaptively migrates NoC workload towards the coolest regions in the chip. The RTM controller uses the DPN to implement the proposed routing strategy.

6.4.1 *Thermal-Adaptive Dynamic Routing in NoCs*

The proposed thermal-adaptive router design is based on the DP and guides the packets to the coolest path among the available paths between a source, s, and a destination, d. The costs of the DP are associated with nodes rather than edges and are equivalent to local temperatures of the cores (T_{local}) that comes from RO-based embedded thermal sensors. At each node there is a DP unit which is connected to other DP units in the system via the DPN. Fig. 6.1 illustrates how the DPN and the sensors are coupled to a 2D mesh NoC. The DPN works as the RTM controller in the proposed scheme.

Assuming a multi-source single destination, each unit receives the costs of the neighbouring units as input, adds its local temperature, and computes and propagates the optimal cost, which is the minimum cost from the local node, c, to the destination node, d, $V^*(c, d)$. By making node cost equal to the local core temperature, the DP chooses the path with a minimum total temperature (i.e. the coolest) among the available shortest paths between a source to a destination. In this scenario, thermal hotspots are avoided whenever possible and the coolest paths are always exploited to minimize the thermal impact of the NoC workload.

Algorithm 6.1 shows the operations taking place in the DP unit to update routing directions in the routing table. The local router's temperature, which is used as the cost by the DP unit, comes from the distributed embedded sensor. These sensors are deployed across

the chip and there is one sensor per core as illustrated in Fig. 6.1. The main algorithm is outlined in lines 1 – 15. Cost updating take

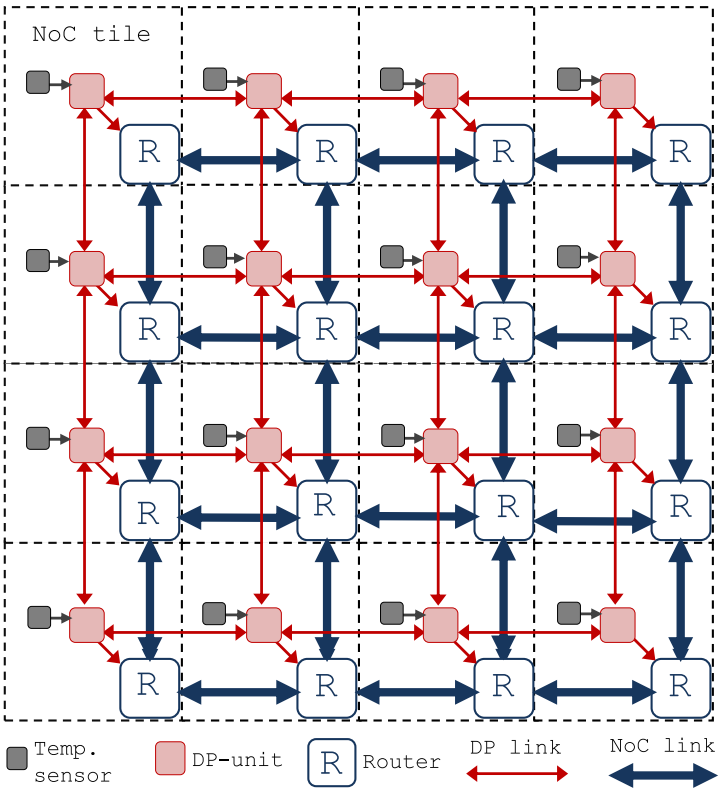


Figure 6.1: Dynamic programming network and temperature sensor array coupled to a 2D mesh NoC.

place for a particular time frame every thermal control cycle. This is controlled by "Compute", which is set to "1" for a time period that is long enough to ensure DPN convergence to the optimal solution. Thus, while Compute = "1", the DP unit executes the DP algorithm (line 1). The DP cost computation starts by checking if the current node is the destination, and if so the DP unit outputs "0" as optimal cost and the routing decision is the local port (lines 2 – 4). For other destinations, the costs are computed in lines 5 – 11. Each DP unit takes the output costs of its direct neighbours as input and computes the cost of reaching the destination through this neighbour. Given a destination d, the expected cost is computed for all *allowable* directions. The minimum cost is then selected, as shown in line 11, and propagated to neighbouring units. This process is repeated until the convergence period ends (Compute="0"). Then the optimal direction $\mu(c, d)$, which is computed as the argument of the minimum operator (in line 13), is used to update the routing table as shown in line 15.

Note that, in the optimal cost and direction computation, only allowable directions are considered. These are the directions returned by the routing function. This is achieved by defining an enable signal

Algorithm 6.1 Pseudo code of the thermal DP unit algorithm.

Define: -
c: current (local) node,
$\mathcal{N}(c)$: all neighbour of node c,
$RT(c, d)$: current node routing table entry for destination, d.
Inputs: -
d: destination node,
$V^*(i, d) : \forall i \in \mathcal{N}(c)$, DP unit output of neighbour node i,
T_{local} : temperature from the local sensor,
$E(i)$: $\forall i \in \mathcal{N}(c)$, is 1 if i is allowable direction, else its 0,
Compute : while 1, the DP unit keeps updating the cost-to-go.
Outputs: -
$\mu(c, d)$: optimal direction from node c to d,
$V^*(c, d)$: optimal cost from node c to d.

while Compute=1 **do**
  if $d = c$ **then**
    $V^*(c, d) = 0$
    $\mu(c, d) = LOCAL$
  else
    for all directions $k \in \mathcal{N}(c, d)$ **do**
      if E(k)=1 **then**
        $V'(c, k, d) = V(k, d) + T_{local}$
      end if
    end for all
    $V^*(c, d) = min_{\forall k|E(k)=1} V'(c, k, d)$
  end if
  $\mu(c, d) = \arg\min_{\forall k|E(k)=1} V'(c, k, d)$
end while
$RT(c, d) \leftarrow \mu(c, d)$ {*update routing table*}

for each direction, i, $E(i)$. This is used in the algorithm to consider only the allowable directions where the corresponding E signal is "1".

The direction enable signal ($E(i)$ for $i \in \{N, S, E, W\}$) comes from the routing function, which ensures deadlock-freeness, as described in the next section. This function takes the coordinates of the current node, c, and destination node, d and return allowable routing directions.

6.4.2 *Deadlock and Livelock Freeness*

The main goal of every routing scheme, whether deterministic or adaptive, is to make sure that packets injected into the network get to their destinations eventually. A packet may not reach its destination for two main reasons: it is involved in either a livelock or a deadlock. Deadlocks are of concern for any adaptive routing algorithm. To guarantee the deadlock-freeness of the proposed routing algorithm, several techniques can be used. For example deadlock detection and recovery can be employed [121]. However, implementing these techniques may lead

to high overheads. For this reason, deadlock avoidance, or turn model, techniques are used in many designs. In this work the negative-first (NF) turn model routing [79] is adopted to ensure deadlock freeness of the proposed thermal-aware adaptive routing. NF routing has a better degree of adaptiveness compared to other types of turn-model routing such as the west-first and north-last. NF turn model prohibits any turns from a positive direction to a negative direction [79].

Livelock means that, as time goes to infinity, the packet continues to circulate in the network without reaching its destination. Livelock can happen only if non-minimal routing is used. The proposed routing algorithm uses only the shortest paths (i.e. the coolest path among the available shortest paths). As a result, the proposed adaptive routing is always free from livelocks.

6.4.3 *Thermal-Aware DP Network Implementation*

The DP network is tightly coupled with the NoC communication fabric as shown in Fig. 6.1. It consists of distributed computational units. At each node, there is a DP unit that implements the DP algorithm and propagates the solution to the neighbouring units. Each computational unit locally exchanges control decisions and other system parameters with the corresponding node.

Fig 6.2 shows the hardware realization of the proposed strategy. The router enables adaptive thermal-aware routing and supports a mesh topology. The routing is table-based and the routing table is updated using the DP unit which implements the proposed strategy. Local temperature sensing provides the DP unit with the local temperature (T_{local}). Also, the routing function provides the direction enable signals to ensure that the DP works in harmony with the routing algorithm (the NF in this work). The costs from four neighbouring nodes (V(N), V(S), V(E) and V(W)) are input to the DP unit and the DP unit propagates the computed optimal cost to those neighbours (as shown at the bottom of Fig. 6.2).

To ensure deadlock-freeness, the routing function circuit shown in Fig. 6.2 implements the negative-first routing algorithm and generates four enable signals, one for each direction. These signals are used by the DP unit to consider only allowable directions when computing the optimal cost and direction. This is achieved by the "direction select" (DS) circuits in the DP unit.

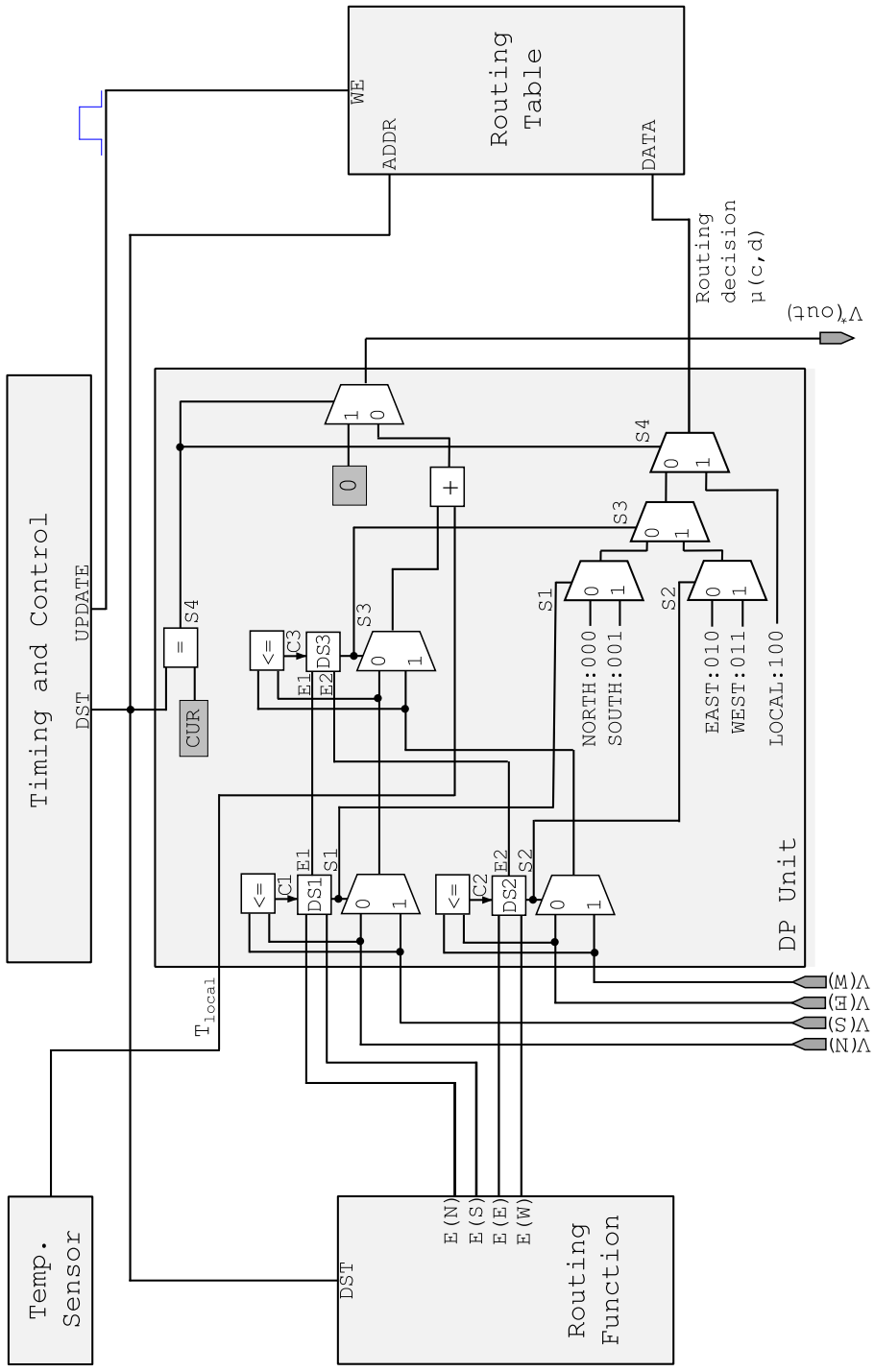


Figure 6.2: Illustration of the hardware implementation of the proposed dynamic thermal-aware routing depicting the updating of the routing table using the DP unit. $V(\mathrm{ch})$ is DP input cost-to-go from channel $\mathrm{ch} \in \{N, S, E, W\}$, $V^*(\mathrm{out})$ is the computed output optimal (minimum) cost, $T_{\mathtt{local}}$ is the local temperature from the sensor, **CUR** is the current (local) address and **DST** is the destination address.

The direction selection circuits (DS) filter directions and, as a result, only allowable directions are returned by the DP unit, enabling it to work in compliance with the deadlock-free routing algorithm. DS circuit make the DPN consider only the directions enabled by the routing function in the optimal DP cost and routing decision computations. The DS circuit has three inputs, E_{in1} and E_{in2} and C_{in} and two outputs, E_{out}, S_{out}. The inputs E_{in1} and E_{in2} are two enable signals and C_{in} is the result of the comparison of DP values. For instance, DS1 in Fig. 6.2 has E(N) and E(S), the two enables of the North and South channels, and C1, the result of the comparison of their costs, V(S) and V(N) as inputs. The outputs are given as follows:

$S_{out} = (C_{in} \wedge E1_{in} \wedge E2_{in}) \vee \overline{E1_{in}}$,

$E_{out} = E1_{in} \vee E2_{in}$.

S_{out} equals to C_{in} only if both directions are enabled ($E_{in1} = 1$ and $E_{in2} = 1$). If only one direction is enabled, S_{out} must select this direction regardless of its cost. The enable signal E_{out} equals to 1 if any of the input directions is enabled. If neither are enabled, E_{out} is 0 and the cost computed at this comparison stage is discarded at the next comparison stage.

6.4.4 *DPN Convergence*

The DPN converges to an optimal routing solution after a delay which is determined by the network topology, the delay of data propagation, and the computational delay of the DP unit. In hardware implementation with parallel execution, each DP unit can simultaneously compute the new expected cost for all neighbouring nodes.

The DPN can be implemented as multiple-destination-multiple-source (MDMS). This means that in each router there are $N = |\mathcal{V}|$ DP units, one for each destination or routing table entry. The total number of DP units for the NoC is N_{dst}^2 (where N_{dst} is the number of destinations). The DPN computes the cost for all destinations in parallel. In this case, the network convergence time is proportional to the network diameter, or the longest path in the network. Thus, the DPN convergence time in clock cycles T_{clk}, will be $N_{diam} \times T_{clk}$. This implementation involves high hardware overheads but this may be necessary with rapidly varying parameters, such as router buffer-level, which can significantly change within a few clock cycles [129].

In this work the parameter used is chip temperature. Compared to clock time period (in the order of nanoseconds), the chip's thermal time constant, τ, is fairly high (in the order of milliseconds or even seconds). Thus, the DPN is implemented as single-destination-multiple-source (SDMS), which implies that for each node there is only one DP unit. This is achieved by a control unit which embeds a destination counter to perform a destination scan once at each thermal control cycle T_{tc} and generates a destination address (DST in Fig. 6.3) to

compute the optimal direction for each destination. All DP units in the DPN must compute the optimal directions for the same destination at the same time. After DPN convergence for the current destination (DST), the "Timing and Control" unit generates the "Update" signal to store the optimal direction for this destination in the routing table.

The control cycle of RTM, T_{tc}, must be less than or equal to τ and the DPN convergence time must be less than T_{tc} to guarantee the proper operation of the DPN as the RTM controller. Thus, in the SDMS case, the following inequality must hold:

$$N_{diam} \times N_{dst} \times T_{clk} < T_{tc}, \tag{6.2}$$

where N_{diam} is the NoC diameter, and N_{dst} is the number of destinations. This implementation requires one DP unit per core which, thus, reduces the hardware cost N times, where N is the total number of nodes in the NoC. As mentioned earlier, inequality 6.2 is guaranteed to hold since $T_{clk} \ll T_{tc}$.

6.4.5 *On-chip Thermal Sensing Implementation*

The proposed RTM system relies on an array of equally-spaced RO-based sensors to extract the spatial distribution of temperature across the chip. The components of these sensors are illustrated in Fig. 6.3. To ensure reliable readings, the output frequency of RO (f_{RO}) must be at least half the sampling frequency (f_{clk}). Since the inverter delay (t_d) is usually much less than the clock period (T_{clk}), bringing f_{RO} down to be less than half f_{clk}, requires an unacceptably large number of inverting elements in the RO. However, a more economical solution to this problem is to use a frequency divider similar to the one shown in Fig. 6.3. A few stages of this divider can bring f_{RO} down to the required range.

The output of the frequency divider is sampled using an edge detector and sampling counter to count the number of oscillations per fixed amount of time. Note that the edge detector output is asynchronous to the sampling clock and cannot be used to drive the sampling counter directly. Thus, the edge detector is followed by at least one flip-flop to synchronize the edge detector output with the sampling clock. The output of the synchronization flip-flop is used to drive the sampling counter.

An array of these sensors is implemented, one sensor per node. Implementing this array of sensors comes with a few challenges of its own. First, ring oscillators are subject to unpredictable process variations and so their temperature models must be calibrated using on-chip readings. Support for this feature has been provided and by choosing to implement the models as lookup tables (as illustrated in the final stage in Fig. 6.3), as opposed to arithmetic logic. This is because such tables can be realized very efficiently using the embedded

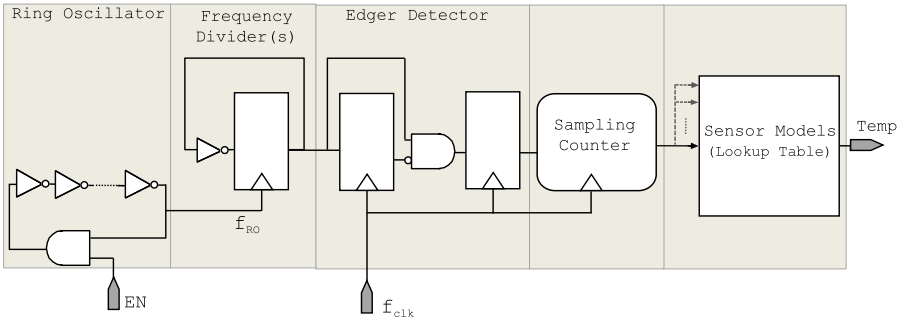


Figure 6.3: Ring oscillator-based thermal sensing components.

M9K memory blocks on the target FPGA device [6]. The only disadvantage of lookup tables is that a sacrifice in precision is involved, since discretization is performed. However, the RO output is susceptible to noise which imposes an upper limit on precision. Thus, discretization is necessary with or without lookup tables.

To generate the sensor models the sensors are calibrated by generating an approximately uniform activity across the FPGA chip by letting uniformly distributed dummy cores to switch with the same frequency. This uniform activity would generate a uniform temperature distribution across the chip. Then the models of the sensors are generated by assuming a linear relationship between the sensor frequency and temperature and taking the counts of the sensors at two temperature points. These points are room temperature, when the cores are not active, and steady state temperature, when all cores are switching. The temperatures are measures using thermocouple that records the temperature and is placed on the top of the package.

There are few sources of errors here. The first is that we are recording the package outer temperature and not the die temperature. To minimize the effect of this source of error we add 5^{o}C to the recorded temperature.

Other source of error is that ring oscillators are highly sensitive to V_{DD} changes. The output of the RO can exhibit spurious output variations corresponding to tens of degrees as a result of nearby switching activity. To overcome this, the RTM controller was designed to initiate each measurement cycle by pausing the system clock during the sensing period to avoid any voltage fluctuations due to switching activity. Thus, sensor readings are subsequently voltage-insensitive temperature readings.

On top of these difficulties, some trade-offs are involved in the choice of measurement period and RTM control cycle duration. Frequent and longer measurements enable the RTM system to adapt more rapidly and accurately to changes in traffic flow. However, extending the time fraction spent in taking temperature measurements also requires that the system clock be paused for longer portions of the NoC's runtime. In the present implementation, it has been found that a measurement

period duration of 5ms and a TM control cycle duration of 1 second can provide fast and effective thermal regulation while keeping the off-time of the NoC within 0.5%.

6.5 RESULTS AND DISCUSSION

To evaluate the proposed thermal control and sensing designs, they were implemented in hardware using an Altera DE2-115 FPGA board equipped with a Cyclone IV E chip. An NoC which consists of 64 routers arranged as an 8×8 mesh is implemented with an input buffer size of 8 flits and a flit size of 32-bits. Table-based routing is used and the routing table entries in the router are updated using its coupled DP unit. The implementation also involves cores that switch in proportion to the local traffic of the router to emulate core switching activity. RO temperature sensors are implemented and deployed as one sensor per core. These sensors are implemented and calibrated as described in Section 6.4.5. The sensors and cores are homogeneously deployed across the FPGA chip area.

The characterization system of the present work is illustrated in Fig 6.4. The Nios II embedded processor [16] is used to perform system configuration and sensor array and routing data collection for experimental analysis. NiosII is also used for the programming of sensor models. A thermocouple works as a calibration reference to calibrate sensor readings and generate sensor model data. The traffic and thermal data are outputted to a PC terminal for analysis and visualisation.

The main system is shown in the dashed box in Fig. 6.4. It consists of the NoC and the cores. The routing decisions of the NoC routers (μ_0 ... μ_{N-1}) are updated by the DP network and the DP network costs are provided by the thermal sensor array. The sensor array counts (C_0 ... C_{N-1}) are converted to the corresponding temperatures (T_0 ... T_{N-1}) using the sensor model logic which uses the M9K memory blocks [15] as lookup tables to implement the models for each sensor in the sensor array.

In the experiments four synthetic traffic patterns are used to evaluate the proposed routing strategy. These are; *Transpose1*, *Transpose2*, *Hotspot*, and *Butterfly*. Assuming $i = 1...X$ and $j = 1...Y$, where X and Y are the x and y dimensions of the NoC, respectively, these traffic patterns are generated as follows: In the *Transpose1* traffic, core(i, j) sends packets to core(j, i). In *Transpose2* traffic, core(i, j) sends packets to core$(X - i, Y - j)$. For *Hotspot* traffic, all tiles send data to the four central cores. Finally, *Butterfly* traffic is generated by letting core$(0, j)$ send to core$(X, Y - j)$.

For the four traffic patterns, the NF routing with DP thermal-aware adaptive selection, which is denoted as DP in the following, is com-

pared with conventional NF routing with buffer-level selection, denoted as BL in the following.

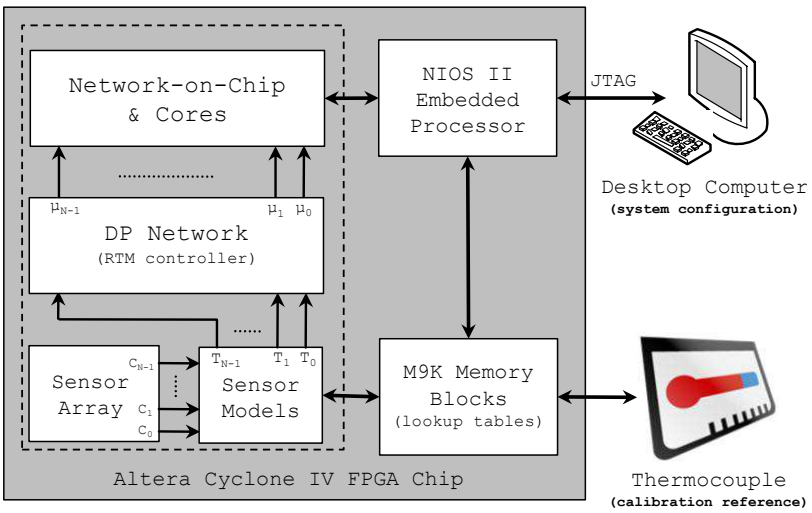
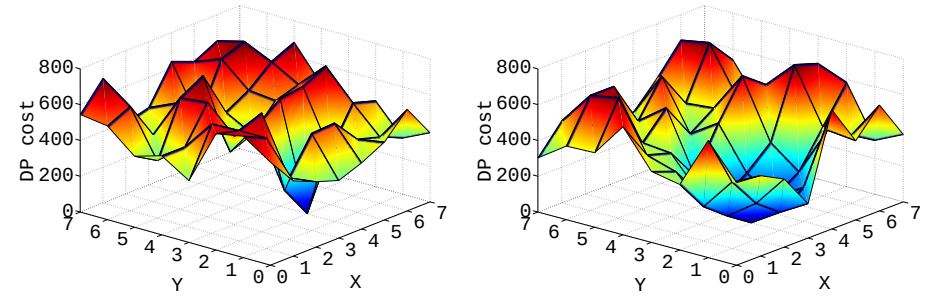


Figure 6.4: Illustration of the FPGA implementation of the NoC thermal characterization system used to obtain the experimental results. μ_i is the routing decisions for node i, C_i is the count output for sensor i, and T_i is the temperature output for sensor i.

6.5.1 *Functional Verification Results*

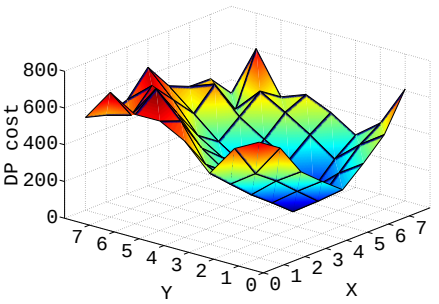
First, only the DPN is implemented in FPGA to verify its proper functionality and to investigate its convergence. The implemented DPN consists of a mesh of 64 DP units. The local costs are set randomly and the DP unit outputs are read after every cycle. Fig. 6.5 illustrate the convergence of the cost-to-go values of all the nodes for destination 20. It can be seen that, by cycle 9 (t=180ns assuming f=50MHz), the DPN converges properly to the optimal cost values for this destination. However, the DPN convergence timer is set to 14 cycles (280ns) since this is the upper limit of convergence for this topology.

Another experiment is conducted to investigate the accuracy of the thermal sensors and to decide on a suitable level of precision in generating the lookup tables of the thermal models. Fig. 6.6 illustrates sensor sampling counter output for a range of temperatures for a 13 inverting stage sensor implemented in FPGA. It is known that, due to their oscillation behaviour, ROs are susceptible to various types of noise that cause variations in propagation delay [180, 72]. However, it can be seen that RO counts accumulate linearly as shown in Fig. 6.6. Based on this, a linear model is adopted to generate the sensor models. For each sensor, 32 counts in a temperature range of $25^oC - 70^oC$ are generated and stored in the M9K memory of the FPGA chip to be used as the lookup table for that sensor. This results in a precision of nearly 1.4^oC for the sensors.

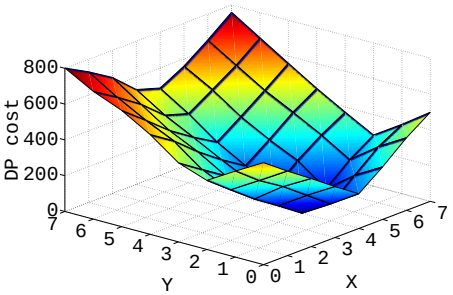


(a) cycle 1

(b) cycle 3



(c) cycle 5



(d) cycle 9

Figure 6.5: Illustration of DPN cost-to-go convergence at different DPN cost computation phases for destination 20.

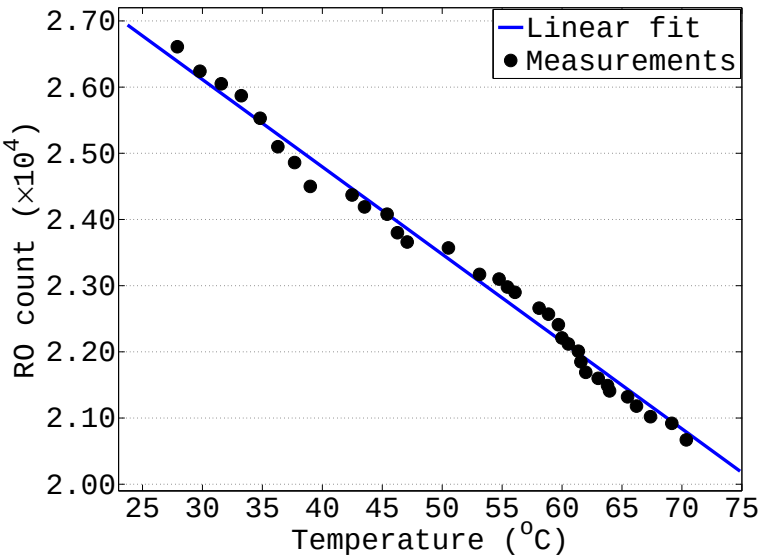
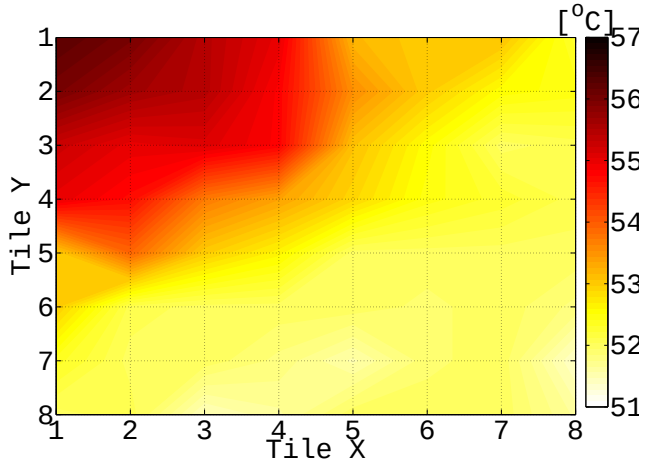
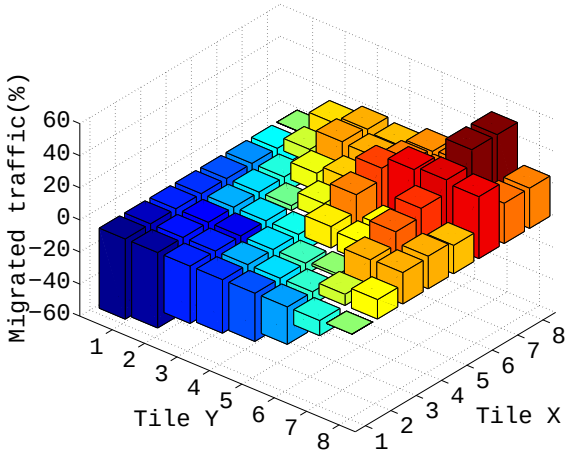


Figure 6.6: Illustration of sensor accuracy.



(a) Temperature gradient created by a hotspot at the chip corner.



(b) Percentage of traffic migrated by the DPN relative to BL routing.

Figure 6.7: Functional verification results: the DPN responds to a chip thermal gradient by migrating traffic to the cooler region in the chip.

Next an experiment is performed to investigate the efficiency of the proposed thermal-aware routing in migrating the traffic in response to thermal gradients in the chip. Here the NoC, DPN and the sensors are all implemented. To create a thermal gradient, a hotspot which consists of 6000 toggling flip-flops is placed in the upper-left corner of the chip. Activating this hotspot creates a significant thermal gradient. The results of this experiment are shown in Fig. 6.7.

Fig. 6.7a shows the steady state thermal gradient created by the hotspot. Taking the BL routing as reference, the percentage of migrated traffic for each of the 64 routers is shown in Fig. 6.7b. The traffic here is *Transpose1*. A negative percentage indicates that traffic is migrated from the router and a positive value indicates that traffic is migrated to the router. It can be seen that the traffic is successfully migrated from the hot region to the cool region and that the migration of traffic

is fully modulated by temperature distribution. This indicates the efficiency of the proposed thermal-adaptive routing in migrating the communication load to cool regions in the chip in response to thermal gradients.

6.5.2 *Spatial Thermal Regulation Results*

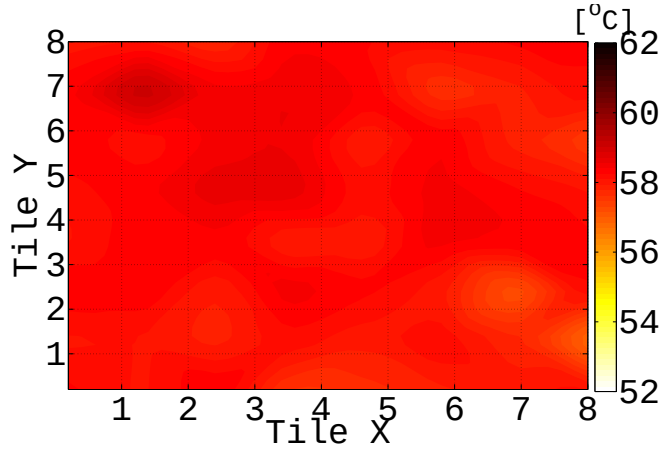
The thermal regulation capability of the DP routing is investigated next. For the four traffic patterns, the DP routing is compared with BL routing in terms of thermal regulation. Packet injection rate for each traffic pattern is calibrated such that both routing techniques have the same throughput for fair comparison.

Each traffic is left to run for 150 seconds, until chip temperature is at equilibrium with ambient temperature. The steady-state spatial thermal distributions for both BL and DP routings for the *Transpose1* traffic are shown in Fig. 6.8. Also, Table 6.1 shows the maxima and ranges of temperature for all the traffic patterns considered with both types of routings.

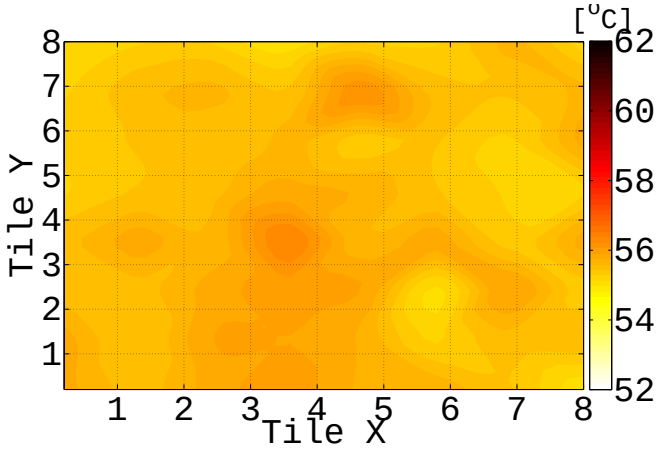
It can be seen that the maximum chip temperature with the DP case is reduced by up to 16.7% compared to BL. Moreover, the chip spatial temperature range in the DP case is reduced by up to 51.7% compared to the BL case. This clearly indicates that the DP creates a more balanced thermal distribution in the chip, which removes thermal hotspots and reduces both the peak and range of chip temperatures. Balanced chip thermal distributions and lower peak temperature would lead to higher reliability and lower power consumption of the chip, since significant thermal gradients can lead to devices wearing out faster and causes higher leakage. Chip performance would also be improved since lower temperatures translate into lower CMOS delay. However, in the following, other important advantage of the proposed RTM is investigated which is slow chip heating and enabling the chip to work longer under a thermal limit.

6.5.3 *Temporal Thermal Regulation Results*

The runtime balancing of chip temperature has another important advantage in addition to reducing the steady-state peak and spatial range of temperature. It also slows down chip heating. Fig. 6.9 plots the maximum chip temperature for both buffer level selection routing (BL) and the proposed DP routing against time for the *Transpose1* traffic case. Also, columns 2 and 3 in Table 6.2 show the thermal time constant (τ) of the chip for all the traffic patterns considered with both DP and BL routings. The thermal time constant (τ) is obtained by



(a)



(b)

Figure 6.8: Example illustrating the chip spatial temperature distributions for *Transpose1* traffic with both a) BL and b) DP routings.

Traffic	Routing				Improvement	
	BL		DP			
	Max	Rng	Max	Rng	Max	Rng
Transpose1	60.0°	3.1°	56.3°	1.5°	16.7%	51.7%
Transpose2	62.9°	4.5°	58.3°	2.2°	16.5%	53.3%
Hotspot	58.8°	3.2°	55.5°	2.1°	13.1%	34.4%
Butterfly	58.1°	2.8°	55.2°	1.5°	12.7%	46.4%

Table 6.1: Results for chip spatial temperature. Maximum and range of temperature for both DP routing and BL routing with percentage improvement in both maximum and range of temperature for the four traffic patterns considered . The improvement is computed after subtracting the initial chip temperature.

fitting the resultant temperature curves to a generic heating model in the form:

$$T(t) = T_0 + \Delta T \times (1 - e^{\frac{-t}{\tau}}) \tag{6.3}$$

where T_0 and T_f are the initial and final temperatures, respectively, and $\Delta T = T_f - T_0$.

It can be seen from the second and third columns of Table 6.2 that τ for the DP case is significantly higher than BL and is nearly double in the case of Transpose1 traffic. This can be attributed to the fact that the DP-driven routing, with global awareness of thermal distribution on the chip, always tends to create a balanced thermal distribution in the chip at runtime. This removes thermal hotspots, reducing both the peak and range of chip temperature and slowing down chip heating.

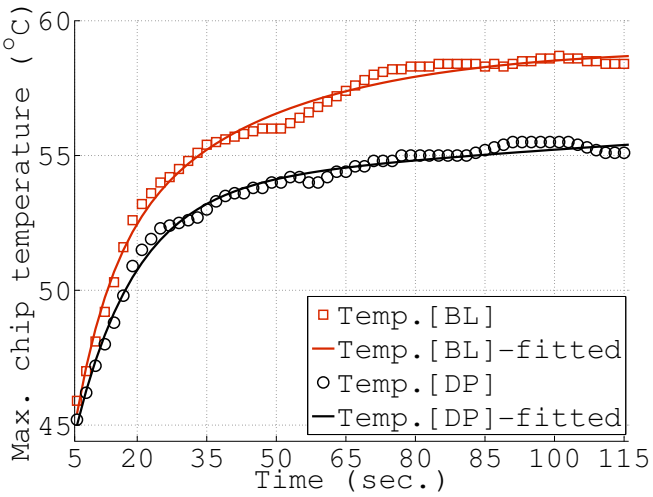


Figure 6.9: Chip heating versus time: maximum temperature for *Transpose1* traffic with both BL routing and thermally adaptive DP routing.

Slower chip heating would enable the chip to work longer before it reaches a thermal limit. In the case of reactive RTM, where throttling is used to react to unacceptable chip heating, slower chip heating postpones throttling and may alleviates it . In this scenario, the chip can deliver higher performance. To illustrate this, the number of delivered packets before the chip maximum temperature reaches a thermal limit of 55°C is evaluated. The results are shown in Table 6.2. It can be noticed that the delaying of thermal limit violation achieved by the DP approach increases the number of packets delivered within this limit by more than double for some types of traffic patterns.

6.5.4 *Performance Evaluation*

Fig. 6.10 compares the performance of the proposed routing strategy (DP) and the buffer-level selection strategy (BL) in terms of average

Traffic	τ (sec.)		#pckts ($\times 10^6$)		DP Imprvmnt	
	BL	DP	BL	DP	τ	#pckts
Transpose1	7.9	15.9	838.6	1728.3	101%	106%
Transpose2	9.3	13.1	697.6	1204.6	41%	72%
Hotspot	11.5	15.8	499.2	999.2	46%	100%
Butterfly	11.8	16.5	952.0	1539.0	40%	62%

Table 6.2: Results of temporal thermal regulation: Thermal time constant, τ, of chip heating and the number of packets delivered within a thermal limit of 55°C for both DP and BL routings with the percentage improvement of DP over BL.

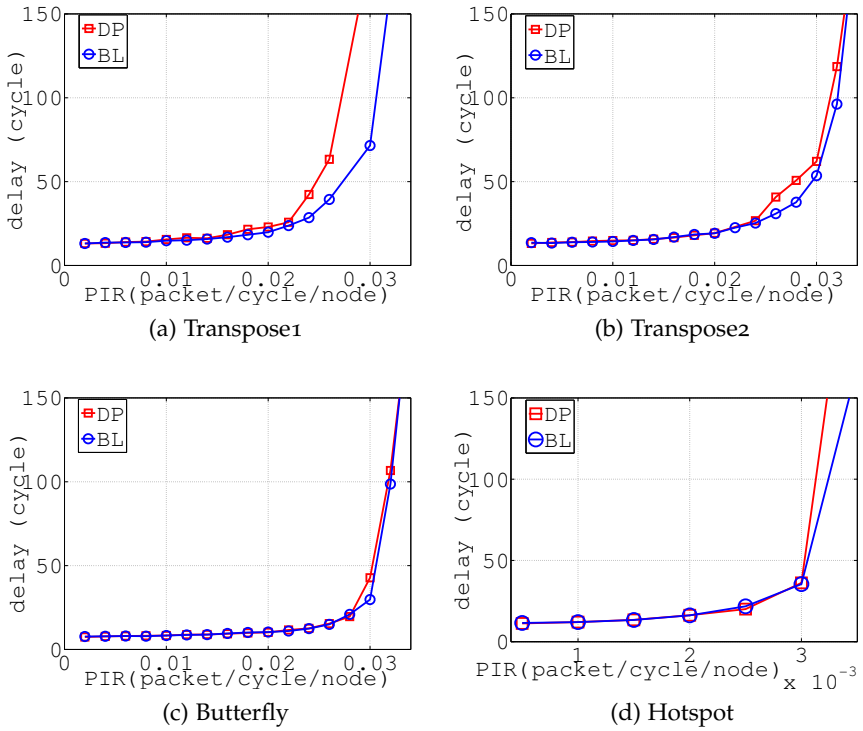
network delay versus the PIR curves under the four traffic scenarios. For the Transpose1 traffic (Fig. 6.10a), it can be seen that there is a slight degradation in the performance of DP compared to BL. However, DP routing does not cause a significant performance overhead compared to the buffer-level selection strategy for most of the traffic scenarios, and this overhead is barely noticeably for the Transpose2, Butterfly and Hotspot traffic patterns.

Component	LUTs	Registers
NoC	96,768	83,648
DPN	205	844
Sensing	4,116	3,052
DPN overhead	0.2%	0.9%
Sensing overhead	4.0%	3.5%
Total overhead (DP+sensing)	4.2%	4.4%

Table 6.3: FPGA implementation results: FPGA resource utilization of DPN and thermal sensing with the percentage overhead relative to 64 core NoC. Sensing hardware include RO sensors, sensor models and lookup tables.

6.5.5 *Hardware Evaluation*

The results of the FPGA implementation of the proposed system are shown in Table 6.3. This table shows the FPGA utilization results in terms of both combinational logic (LUTs) and register utilizations of each component in the implemented system. The ROs, sensor models and lookup tables are considered to be part of the thermal sensing



(a) Transpose1 (b) Transpose2

(c) Butterfly (d) Hotspot

Figure 6.10: Performance curves in terms of delay versus packet injection rate (PIR) for both DP and BL.

component. It can be seen that the total overhead of both the thermal sensing and control is about 4%. It can also be noticed that the DP network has a very low overhead compared to the NoC (less than 1%) and the majority of the overhead is due to the sensing component. It's worth mentioning here that this implementation of the sensing focused mainly on accuracy rather than minimizing the overhead. Thus, it is believed that, with more effort, the sensing overhead can also be reduced. Moreover, some current NoC prototypes (e.g. Intel's SCC [161]) have embedded thermal sensors that can be used with the proposed technique without any thermal sensing overhead at all.

6.6 SUMMARY AND CONCLUSION

A thermal-adaptive routing design and implementation for NoCs is presented in this chapter. A dynamic programming-based routing control architecture is proposed. The DPN is used to update routing decisions with awareness of intra-chip thermal variations. ROs are used for temperature sensing to capture these variations. Many design and implementation challenges associated with sensor accuracy and precision, as well as DPN implementation, are addressed. The proposed system is implemented and evaluated in FPGA. The results show that the proposed routing responds efficiently to thermal variations in

the chip and can significantly reduce peak chip temperatures and spatial thermal gradients. The proposed routing strategy can also slow down chip heating due to minimized hotspot formation and thermal moderation. Moreover, low hardware and performance overheads are introduced by the proposed routing. For future many core systems, where the communication budget is subject to considerable increase, the proposed technique would lead to higher chip reliability. Moreover, it would improve performance when throttling is employed to react to unacceptable thermal states.

7 CONCLUSIONS AND FUTURE WORK

7.1 SUMMARY AND CONCLUSION

The shrinking of feature size is enabling the integration of higher numbers of cores in a chip, and silicon density is continuously increasing. One consequence of this increasing density is a change in on-chip system design from being computation centric to communication-centric. NoCs are proposed to provide the scalability, reliability and power efficiency needed for inter-chip communication in these high density CMPs and MPSoCs. Another consequence of the higher scale of integration is the increasing difficulty in ensuring the integrity of the chip's physical parameter due to higher power density and operating frequencies as well as process variations. In particular, power and thermal integrity, which are crucial for VLSI system reliability and performance, are becoming increasingly difficult to guarantee due to higher power density and operating frequencies which result in increased temperature and greater voltage drops. As a result, addressing the power and thermal integrity challenges for future many-core systems at one level of abstraction would not be sufficient to guarantee physical parameter integrity. Design time and runtime strategies, at various levels, may need to work together to provide this integrity in large many-core systems. This thesis advocates strategies that work at the level of the on-chip network, with its rising power budget, in order to improve power and thermal integrity. This section summarizes the study and presents its main conclusions.

Power integrity has become a critical concern with the rapid shrinking of feature size and the ever-increasing power consumption associated with nanometre-scale integration. In particular, on-chip communication in NoC platforms dictates the power dissipation in and overall performance of multi-core systems. These architectures require a dedicated model for analysing intra-chip power supply variations, which must embed distinctive communication characteristics and parameters. This thesis studies power supply variations in NoC-based systems. A dedicated model for determining on-chip V_{DD} drops is proposed, discussed and verified. Then, this model is used to investigate power supply variations caused by various traffic patterns and routing algorithms in NoCs. Moreover, the impact of the modelled V_{DD} variations and PSN on NoC links is investigated. This is conducted via a statistical timing analysis of NoC links in the presence of PSN which lead to the evaluation of the resulting timing violations and BERs. It has been observed that unbalanced NoC workloads can be caused by unbalanced

traffic patterns, routing algorithms or application mappings, and these can cause significant intra-chip voltage variations. It has also been observed that higher variations can also be reflected in higher rates timing of violations and BERs.

It is concluded from these analyses of power supply variations in NoCs that PSN strongly correlates with the spatial distribution of activity densities. This thesis defines a metric for regional activity density and studies the impact of various patterns of this metric on PSN. Particular patterns of activity density are found to have higher correlations with PSN. Thus, this thesis also proposes a new application mapping strategy in NoCs that aims to create balanced activity across the chip by employing a force-based optimization. Evaluation using many real-application benchmarks shows a significant reduction in PSN and resulting bit error rates, with negligible performance and energy penalties. Moreover, it has been observed that the PSN reduction achieved by the proposed mapping is consistent for smaller technology nodes. The proposed mapping strategy would improve power integrity and reliability in future large many-core systems.

Thermal integrity, on the other hand, is of concern with any VLSI system. However, due to substantial silicon density, this challenge is more prominent in 3D VLSI. Despite its numerous advantages, 3D integration introduces serious thermal threats that may increase faults and system failures. Particularly, in three-dimensional network-on-chip (3D NoC) systems integration, without proper thermal dispersion, could lead to ultra-high temperature hotspots, increasing soft errors and the leakage of power and subsequently reduced device reliability and lifespan. This thesis introduces an adaptive runtime strategy to effectively optimize heat distribution in the 3D geometry. This strategy employs the DPN to select and optimize the direction of data manoeuvring. Existing 2D routing algorithms are extended to 3D and improved to increase path diversity. The proposed routing strategy is compared with recent thermal optimization techniques. The new approach is found to achieve better thermal moderation and reliability, and higher throughput performance. It has been observed that maximum temperature can be reduced by up to 18°C and that temperature gradient, reflecting spatial variation, is reduced by more than half compared to other methods. This thermal mitigation leads to a 63% reliability improvement. Moreover, given a thermal limit, the throughput is improved by more than 23%.

This thesis also presents a detailed design and hardware implementation of this dynamic thermal-adaptive routing strategy. The DPN implements the adaptive routing control logic and ROs are used for temperature sensing. Various implementation and design choices associated with the DPN and the sensors are presented. This includes DPN convergence analysis. Also, sensor accuracy and precision issues, such as isolating the IR drops and intra-chip process variations, are ad-

dressed. Implementing the proposed routing strategy in FPGA shows promising results in terms of functionality and thermal regulation with a variety of traffic patterns. In terms of functionality, the proposed scheme is shown to be highly flexible in manoeuvring packets away from hot regions. A reduction of 16% in the maximum chip temperature and a thermal gradient reduction of 51% compared with performance-driven routing are observed. Another important advantage of the proposed scheme is that it slows down chip heating. This slower chip heating is shown to be reflected in up to 100% higher performance when the chip works under a thermal limit.

7.2 FUTURE WORK

Many research directions can be followed based on the techniques, algorithms and architectures propose in this thesis. Some recommendations for extending the works in this thesis are presented in this section.

The proposed mapping strategy can be extended by introducing task dependency in the application model. The activity density and, consequently, forces can be a function of time as well as space. Thus, the activity density can be expressed as a function of time in addition to region size. In this scenario, the proposed technique can be integrated in a unified scheduling/mapping algorithm to achieve both temporal and spatial activity density minimization. This problem will be looked into in future developments of the proposed new mapping strategy.

The thermal-adaptive strategy can be extended such that the thermal dynamics of the application are considered. For instance, thermal control cycles can be variable depending on the application's thermal dynamics. Also, various techniques can be explored for DPN implementation. For example, hierarchical DPN topologies can be implemented and evaluated to reduce DPN convergence time. The runtime reconfigurability of the DPN could be used, which mean that the DPN can change the topology to adapt to dynamic task allocation at runtime and power management techniques, such as power gating [197], or dark silicon [69] in many-core NoC systems.

Moreover, other physical parameters can also be explored. One important parameter is process variation which is exacerbated by technology scaling and may cause significant deviations in post-silicon chip performance compared to the original design. Thus, equipping NoCs with static or dynamic adaptivity to these variations is a rich area of research.

Part II

Thesis Appendices

A

THERMAL MATERIAL PARAMETERS

Table A.1 gives the parameter settings of the die layers and IC package layers used in Chapter 5 for thermal simulation. These parameters are used with *HotSpot* thermal modelling tool for NoC dynamic thermal simulation to compute the temperature distribution at different parts in the chip.

Table A.1: The material parameters used in this work for the thermal simulation.

IC layer/material	Thermal Conductivity (W/(m.K))	Thickness (m)	Width (m)	Length (m)
Heatsink/copper	400	0.0069	0.06	0.06
Heat spreader/copper	400	0.001	0.03	0.03
Thermal interface material	4	2e-5	0.009	0.012
Die –layer 0/silicon	100	0.0015	0.009	0.012
Die –layer 1/silicon	100	0.0015	0.009	0.012
Die –layer 2/silicon	100	0.0015	0.009	0.012
Die –layer 3/silicon	100	0.0015	0.009	0.012
Metal layers (8)/copper	400	0.00001	0.009	0.012
Underfill	1.25	0.00001	0.009	0.012
C4 pads	2.5	0.00001	0.000001	0.000001
Chip carrier (ceramic)	2	0.001	0.021	0.021
Solder balls	16.6	0.00094	0.021	0.021
Printed circuit board	3	0.002	0.1	0.1

B

DETAILS OF POWER DELIVERY PARAMETERS AND TECHNOLOGY SCALING

B.1 SETUP

Table B.1 summarizes the experimental settings for power supply noise computation in Chapter 4.

Table B.1: Experimental setup and parameters.

Parameter	Setting
On-chip PDN granularity	5×5 per NoC tile
Tile dimensions	$2mm \times 1.5mm$
Tile Floorplan	Intel's TeraFlop [193]
Off-chip PDN parameters	From Gupta et al [85]
On-chip PDN parameters	From PTM [4]
NoC Topology	Mesh
NoC Flit Size	39 bits
NoC Buffer Size	16 flits
NoC Packet Size	3 flits
V_{DD}	1 v
Frequency	3 GHz

B.2 SCALING PARAMETERS

Table B.1 summarizes the experimental settings for power supply noise computation in Chapter 4.

Table B.2: Technology scaling factors for various parameters.

	Technology generation				
	65 nm	45 nm	32 nm	22 nm	18 nm
V_{DD}	1.0	0.8	0.7	0.6	0.6
Area	1.0	0.7	0.6	0.4	0.3
Frequency	1.0	1.2	1.4	1.7	2.1
Energy	1.0	1.1	1.3	1.7	2.4

Part III

Thesis Bibliography

BIBLIOGRAPHY

[1] BookSim: Interconnection Network Simulator. URL http://nocs.stanford.edu/cgi-bin/trac.cgi/wiki/Resources/BookSim#. [Dec 02, 2013].

[2] CONNECT home page. URL http://http://users.ece.cmu.edu/~mpapamic/connect/. [Nov. 30, 2013].

[3] Nostrum NoC home page. URL http://www.ict.kth.se/nostrum/. [Nov. 30, 2013].

[4] PTM: Predictive technology model. http://ptm.asu.edu/.

[5] SpiNNaker home page. URL http://apt.cs.man.ac.uk/projects/SpiNNaker/. [Nov. 27, 2013].

[6] Internal Memory (RAM and ROM), User Guide, May 2013. URL http://www.altera.co.uk/literature/ug/ug_ram_rom.pdf.

[7] C. Ababei, H. S. Kia, O. P. Yadav, and Jingcao Hu. Energy and reliability oriented mapping for regular networks-on-chip. In *Networks on Chip (NoCS), 2011 Fifth IEEE/ACM International Symposium on*, pages 121–128, 2011.

[8] C. Addo-Quaye. Thermal-aware mapping and placement for 3-d noc designs. In *SOC Conference, 2005. Proceedings. IEEE International*, pages 25 –28, sept. 2005. doi: 10.1109/SOCC.2005.1554447.

[9] A. Adriahantenaina, H. Charlery, A. Greiner, L. Mortiez, and C.A. Zeferino. Spin: a scalable, packet switched, on-chip micro-network. In *Design, Automation and Test in Europe Conference and Exhibition, 2003*, pages 70–73 suppl., 2003. doi: 10.1109/DATE.2003.1253808.

[10] A. Agarwal, K. Chopra, D. Blaauw, and V. Zolotov. Circuit optimization using statistical static timing analysis. In *Design Automation Conference '05. Proceedings. 42nd*, pages 321 – 324, june 2005. doi: 10.1109/DAC.2005.193825.

[11] Amir H. Ajami, Kaustav Banerjee, and Massoud Pedram. Scaling analysis of on-chip power grid voltage variations in nanometer scale ulsi. *Analog Integrated Circuits and Signal Processing*, 42(3): 277–290, 2005.

[12] R. Al-Dujaily, T. Mak, Kuan Zhou, Kai-Pui Lam, Yicong Meng, A. Yakovlev, and Chi-Sang Poon. On-chip dynamic programming networks using 3D-tsv integration. In *Embedded Computer*

Systems (SAMOS), 2011 International Conference on, pages 318 –325, july 2011. doi: 10.1109/SAMOS.2011.6045478.

[13] R. Al-Dujaily, T. Mak, Fei Xia, A. Yakovlev, and M. Palesi. Embedded transitive closure network for runtime deadlock detection in networks-on-chip. *Parallel and Distributed Systems, IEEE Transactions on*, 23(7):1205–1215, 2012. ISSN 1045-9219. doi: 10.1109/TPDS.2011.275.

[14] Ra'ed Al-Dujaily, Nizar Dahir, Terrence Mak, Fei Xia, and Alex Yakovlev. Dynamic programming-based runtime thermal management (dprtm): An online thermal control strategy for 3d-noc systems. *ACM Trans. Des. Autom. Electron. Syst.*, 19(1):2:1–2:27, December 2013. ISSN 1084-4309. doi: 10.1145/2534382. URL http://doi.acm.org/10.1145/2534382.

[15] Altera. Cyclone IV Device Handbook, 2013. URL http://www.altera.co.uk/literature/hb/nios2/n2cpu_nii5v1.pdf.

[16] Altera. Nios II Processor Reference Handbook, 2011. URL http://www.altera.co.uk/literature/hb/nios2/n2cpu_nii5v1.pdf.

[17] K. Arabi, R. Saleh, and Meng Xiongfei. Power supply noise in socs: Metrics, management, and measurement. *Design Test of Computers, IEEE*, 24(3):236 –244, may-june 2007. ISSN 0740-7475. doi: 10.1109/MDT.2007.79.

[18] G. Ascia, V. Catania, and M. Palesi. Multi-objective mapping for mesh-based noc architectures. In *Hardware/Software Codesign and System Synthesis, 2004. CODES + ISSS 2004. International Conference on*, pages 182 – 187, sept. 2004. doi: 10.1109/CODESS.2004.241215.

[19] Giuseppe Ascia, Vincenzo Catania, and Maurizio Palesi. Mapping cores on network-on-chip. *International Journal of Computational Intelligence Research*, 1(1):109–126, 2005.

[20] D. Atienza, P.G. Del Valle, G. Paci, F. Poletti, L. Benini, G. De Micheli, and J.M. Mendias. A fast hw/sw fpga-based thermal emulation framework for multi-processor system-on-chip. In *Design Automation Conference, 2006 43rd ACM/IEEE*, pages 618–623, 2006. doi: 10.1109/DAC.2006.229307.

[21] M. Bahmani, A. Sheibanyrad, F. Petrot, F. Dubois, and P. Durante. A 3d-noc router implementation exploiting vertically-partially-connected topologies. In *VLSI (ISVLSI), 2012 IEEE Computer Society Annual Symposium on*, pages 9–14, 2012. doi: 10.1109/ISVLSI.2012.19.

[22] G. Bai, S. Bobba, and I.N. Hjj. Static timing analysis including power supply noise effect on propagation delay in vlsi circuits. In *Design Automation Conference, 2001. Proceedings*, pages 295 – 300, 2001. doi: 10.1109/DAC.2001.156154.

[23] L. Benini and G. De Micheli. Networks on chips: a new SoC paradigm. *IEEE Computer*, 35(1):70 –78, 2002. ISSN 0018-9162. doi: 10.1109/2.976921.

[24] E. Beyne. 3D system integration technologies. In *VLSI Technology, Systems, and Applications, 2006 International Symposium on*, pages 1 –9, april 2006. doi: 10.1109/VTSA.2006.251113.

[25] Sanjukta Bhanja and N Ranganathan. Switching activity estimation of vlsi circuits using bayesian networks. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 11(4):558–567, 2003.

[26] T. Bjerregaard. *The MANGO clockless network-on-chip: Concepts and implementation*. PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2005. URL http://www2.imm.dtu.dk/pubdb/p.php?4025. Supervised by Assoc. Prof. Jens Sparsø, IMM.

[27] Tobias Bjerregaard and Shankar Mahadevan. A survey of research and practices of network-on-chip. *ACM Comput. Surv.*, 38(1), June 2006. ISSN 0360-0300. doi: http://doi.acm.org/10.1145/1132952.1132953. URL http://doi.acm.org/http://doi.acm.org/10.1145/1132952.1132953.

[28] S. Bodapati and F. N. Najm. High-level current macro-model for power-grid analysis. In *39th Design Automation Conference Proceedings.*, pages 385–390, 2002.

[29] P. Bogdan and R. Marculescu. Non-stationary traffic analysis and its implications on multicore platform design. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 30(4):508 –519, april 2011. ISSN 0278-0070. doi: 10.1109/TCAD.2011.2111270.

[30] Paul Bogdan and Radu Marculescu. Statistical physics approaches for network-on-chip traffic characterization. In *Proceedings of the 7th IEEE/ACM international conference on Hardware/-software codesign and system synthesis*, CODES+ISSS '09, pages 461–470, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-628-1. doi: 10.1145/1629435.1629498. URL http://doi.acm.org/10.1145/1629435.1629498.

[31] B. Boghrati and S. Sapatnekar. A scaled random walk solver for fast power grid analysis. In *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pages 1–6, 2011.

[32] S. Borkar. 3d integration for energy efficient system design. In *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, pages 214–219, 2011.

[33] Shekhar Borkar. Thousand core chips: a technology perspective. In *Proceedings of the 44th annual Design Automation Conference*, DAC '07, pages 746–749, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-627-1. doi: 10.1145/1278480.1278667. URL http://doi.acm.org/10.1145/1278480.1278667.

[34] Yan Boyuan, S. X. D. Tan, Chen Gengsheng, and Wu Lifeng. Modeling and simulation for on-chip power grid networks by locally dominant krylov subspace method. In *Computer-Aided Design ICCAD. IEEE/ACM International Conference on*, pages 744–749, 2008.

[35] David Brooks, Robert P. Dick, Russ Joseph, and Li Shang. Power, thermal, and reliability modeling in nanometer-scale microprocessors. *IEEE Micro*, 27:49–62, 2007. ISSN 0272-1732. doi: http://doi.ieeecomputersociety.org/10.1109/MM.2007.58.

[36] Y. Cao, T. Sato, D. Sylvester, M. Orshansky, and C. Hu. Predictive technology model. *Nanoscale integration and modeling group, Arizona State Univerity, http://ptm.asu.edu/*, 2006.

[37] M.F. Chang and *et al.* Cmp network-on-chip overlaid with multi-band rf-interconnect. In *High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on*, pages 191 –202, feb. 2008. doi: 10.1109/HPCA.2008.4658639.

[38] Chih-Hao Chao, Kai-Yuan Jheng, Hao-Yu Wang, Jia-Cheng Wu, and An-Yeu Wu. Traffic- and thermal-aware run-time thermal management scheme for 3D noc systems. In *Networks-on-Chip (NOCS), 2010 Fourth ACM/IEEE International Symposium on*, pages 223 –230, may 2010. doi: 10.1109/NOCS.2010.32.

[39] H. H. Chen and D. D. Ling. Power supply noise analysis methodology for deep-submicron VLSI chip design. In *Design Automation Conference, 1997. Proceedings of the 34th*, pages 638–643, 1997.

[40] H. H. Chen and J. S. Neely. Interconnect and circuit modeling techniques for full-chip power supply noise analysis. *Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging, IEEE Transactions on*, 21(3):209–215, 1998.

[41] J. Chen and L. He. A decoupling method for analysis of coupled rlc interconnects. In *Proc. IEEE/ACMInt. Great Lakes Symp. VLSI,*,

pages 41–46. IEEE, 2002. Proceedings of the 12th ACM Great Lakes symposium on VLSI.

[42] Kun-Chih Chen, Chih-Hao Chao, Shu-Yen Lin, and An-Yeu(Andy) Wu. Traffic- and thermal-aware routing algorithms for 3d network-on-chip (3d noc) systems. In Maurizio Palesi and Masoud Daneshtalab, editors, *Routing Algorithms in Networks-on-Chip*, pages 307–338. Springer New York, 2014. ISBN 978-1-4614-8273-4. doi: 10.1007/978-1-4614-8274-1_12. URL http://dx.doi.org/10.1007/978-1-4614-8274-1_12.

[43] L. H. Chen, M. Marek-Sadowska, and F. Brewer. Coping with buffer delay change due to power and ground noise. In *Design Automation Conference '02. Proceedings. 39th*, pages 860–865, 2002.

[44] Ge-Ming Chiu. The odd-even turn model for adaptive routing. *Parallel and Distributed Systems, IEEE Transactions on*, 11(7):729 –738, jul 2000. ISSN 1045-9219. doi: 10.1109/71.877831.

[45] Jinseong Choi, M. Swaminathan, Nhon Do, and R. Master. Modeling of power supply noise in large chips using the circuit-based finite-difference time-domain method. *Electromagnetic Compatibility, IEEE Transactions on*, 47(3):424 – 439, aug. 2005. ISSN 0018-9375. doi: 10.1109/TEMC.2005.851719.

[46] Chen-Ling Chou and R. Marculescu. User-aware dynamic task allocation in networks-on-chip. In *Design, Automation and Test in Europe, 2008. DATE '08*, pages 1232–1237, 2008. doi: 10.1109/DATE.2008.4484847.

[47] J. Cong, Jie Wei, and Yan Zhang. A thermal-driven floorplanning algorithm for 3d ics. In *Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference on*, pages 306–313, Nov 2004. doi: 10.1109/ICCAD.2004.1382591.

[48] A. R. Conn, R. A. Haring, and C. Visweswariah. Noise considerations in circuit optimization. pages 220–227. ACM, 1998. Proceedings of the 1998 IEEE/ACM international conference on Computer-aided design.

[49] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press and McGraw-Hill Publishers, USA, 2001.

[50] A.K. Coskun, T.S. Rosing, and K. Whisnant. Temperature aware task scheduling in mpsocs. In *Design, Automation Test in Europe Conference Exhibition, 2007. DATE '07*, pages 1–6, April. doi: 10.1109/DATE.2007.364540.

[51] Nizar Dahir, Terrence Mak, and Alex Yakovlev. Communication centric on-chip power grid models for networks-on-chip. In *VLSI*

and System-on-Chip (VLSI-SoC), 2011 IEEE/IFIP 19th International Conference on, pages 180–183, 2011.

[52] Nizar Dahir, Ra'ed Al-Dujaily, Alex Yakovlev, Petros Missailidis, and Terrence Mak. Deadlock-free and plane-balanced adaptive routing for 3D networks-on-chip. In *Proceedings of the Fifth International Workshop on Network on Chip Architectures*, NoCArc '12, pages 31–36, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1540-1. doi: 10.1145/2401716.2401724. URL http://doi.acm.org/10.1145/2401716.2401724.

[53] Nizar Dahir, Terrence Mak, Fei Xia, and Alex Yakovlev. Minimizing power supply noise through harmonic mappings in networks-on-chip. In *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, CODES+ISSS '12, pages 113–122, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1426-8. doi: 10.1145/2380445.2380468. URL http://doi.acm.org/10.1145/2380445.2380468.

[54] Nizar Dahir, Terrence Mak, Fei Xia, and Alex Yakovlev. Modelling and tools for power supply variations analysis in networks-on-chip. *IEEE Transactions on Computers*, PP(99):1, 2012. ISSN 0018-9340. doi: http://doi.ieeecomputersociety.org/10.1109/TC.2012.272.

[55] Nizar Dahir, Ra'ed Al-Dujaily, Terrence Mak, and Alex Yakovlev. Thermal Optimization in Network-on-Chip Based 3D Chip Multiprocessors Using Dynamic Programming Networks. *ACM Transactions on Embedded Computing Systems*, (Review Submitted), 2013.

[56] Nizar Dahir, Terrence Mak, Ra'ed Al-Dujaily, and Alex Yakovlev. Highly adaptive and deadlock-free routing for three-dimensional networks-on-chip. *Institution of Engineering and Technology*, 7:255–263, November 2013. URL http://digital-library.theiet.org/content/journals/10.1049/iet-cdt.2013.0029.

[57] Nizar Dahir, Ghaith Tarawneh, Ra'ed Al-Dujaily, Terrence Mak, and Alex Yakovlev. Design and Implementation of Dynamic Thermally-Adaptive Routing Strategy for Networks-on-Chip. In *Parallel, Distributed and Network-Based Processing (PDP), 2014 22nd Euromicro International Conference on*, volume (Submitted), pages 141–144, 2014.

[58] M. Dall'Osso, G. Biccari, L. Giovannini, D. Bertozzi, and L. Benini. Xpipes: a latency insensitive parameterized network-on-chip architecture for multiprocessor socs. In *Computer Design, 2003. Proceedings. 21st International Conference on*, pages 536–539, 2003. doi: 10.1109/ICCD.2003.1240952.

[59] W. J. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers, USA, 2004.

[60] William J. Dally. Virtual-channel flow control. In *Proceedings of the 17th annual international symposium on Computer Architecture*, ISCA '90, pages 60–68, New York, NY, USA, 1990. ACM. ISBN 0-89791-366-3. doi: 10.1145/325164.325115. URL http://doi.acm.org/10.1145/325164.325115.

[61] William J. Dally and Hiromichi Aoki. Deadlock-free adaptive routing in multicomputer networks using virtual channels. *Parallel and Distributed Systems, IEEE Transactions on*, 4(4):466–475, 1993.

[62] W.J. Dally and C.L. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. *Computers, IEEE Transactions on*, C-36(5):547 –553, may 1987. ISSN 0018-9340. doi: 10.1109/TC.1987.1676939.

[63] W.J. Dally and B. Towles. Route packets, not wires: on-chip interconnection networks. In *DAC-2001*, pages 684 – 689, 2001.

[64] M. Daneshtalab. *Exploring Adaptive Implementation of On-Chip Networks*. PhD thesis, Department of Information Technology, University of Turku, Finland, 2011.

[65] J. Donald and M. Martonosi. Techniques for multicore thermal management: Classification and new exploration. In *Computer Architecture, 2006. ISCA '06. 33rd International Symposium on*, pages 78 –88, 0-0 2006. doi: 10.1109/ISCA.2006.39.

[66] Jose Duato, Sudhakar Yalamanchili, and Ni Lionel. *Interconnection Networks: An Engineering Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002. ISBN 1558608524.

[67] Victor Dumitriu and Gul N Khan. Throughput-oriented noc topology generation and analysis for high performance socs. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 17(10):1433–1446, 2009.

[68] Hans Eisenmann and F.M. Johannes. Generic global placement and floorplanning. In *Design Automation Conference, 1998. Proceedings*, pages 269–274, June 1998.

[69] H. Esmaeilzadeh, E. Blem, R. St.Amant, K. Sankaralingam, and D. Burger. Dark silicon and the end of multicore scaling. In *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*, pages 365–376, 2011.

[70] F. Fazzino, M. Palesi, and D. Patti. Noxim: Network-on-chip simulator. *URL: http://sourceforge.net/projects/noxim*, 2008.

[71] B.S. Feero and P.P. Pande. Networks-on-chip in a three-dimensional environment: A performance evaluation. *Computers, IEEE Transactions on*, 58(1):32 –45, jan. 2009. ISSN 0018-9340. doi: 10.1109/TC.2008.142.

[72] J.J.L. Franco, E. Boemo, E. Castillo, and L. Parrilla. Ring oscillators as thermal sensors in fpgas: Experiments in low voltage. In *Programmable Logic Conference (SPL), 2010 VI Southern*, pages 133–137, 2010. doi: 10.1109/SPL.2010.5483027.

[73] Binzhang Fu, Yinhe Han, Huawei Li, and Xiaowei Li. The abacus turn model. In Maurizio Palesi and Masoud Daneshtalab, editors, *Routing Algorithms in Networks-on-Chip*, pages 69–103. Springer New York, 2014. ISBN 978-1-4614-8273-4. doi: 10.1007/978-1-4614-8274-1_4. URL http://dx.doi.org/10.1007/978-1-4614-8274-1_4.

[74] Stephen Furber and Andrew Brown. Biologically-inspired massively-parallel architectures-computing beyond a million processors. In *Application of Concurrency to System Design, 2009. ACSD'09. Ninth International Conference on*, pages 3–12. IEEE, 2009.

[75] Yang Ge, Parth Malani, and Qinru Qiu. Distributed task migration for thermal management in many-core systems. In *Proceedings of the 47th Design Automation Conference*, DAC '10, pages 579–584, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0002-5. doi: 10.1145/1837274.1837417. URL http://doi.acm.org/10.1145/1837274.1837417.

[76] Abhijit Ghosh, Srinivas Devadas, Kurt Keutzer, and Jacob White. Estimation of average switching activity in combinational and sequential circuits. In *Proceedings of the 29th ACM/IEEE Design Automation Conference*, pages 253–259. IEEE Computer Society Press, 1992.

[77] P. Ghosh, A. Sen, and A. Hall. Energy efficient application mapping to noc processing elements operating at multiple voltage levels. In *Networks-on-Chip, 2009. NoCS 2009. 3rd ACM/IEEE International Symposium on*, pages 80–85, 2009. doi: 10.1109/NOCS.2009.5071448.

[78] Ascia Giuseppe, Catania Vincenzo, Palesi Maurizio, and Patti Davide. Neighbors-on-path: A new selection strategy for on-chip networks. In *Embedded Systems for Real Time Multimedia, Proceedings of the IEEE/ACM/IFIP Workshop on*, pages 79–84, 2006.

[79] C.J. Glass and L.M. Ni. The turn model for adaptive routing. In *Computer Architecture, 1992. Proceedings., The 19th An-*

nual International Symposium on, pages 278 –287, 1992. doi: 10.1109/ISCA.1992.753324.

[80] K. Goossens, J. Dielissen, O.P. Gangwal, S.G. Pestana, A. Radulescu, and E. Rijpkema. A design flow for application-specific networks on chip with guaranteed performance to accelerate soc design and verification. In *Design, Automation and Test in Europe, 2005. Proceedings*, pages 1182–1187, 2005. doi: 10.1109/DATE.2005.11.

[81] K. Goossens, J. Dielissen, and A. Radulescu. Aethereal network on chip: concepts, architectures, and implementations. *Design Test of Computers, IEEE*, 22(5):414–421, 2005. ISSN 0740-7475. doi: 10.1109/MDT.2005.99.

[82] A. Goyal and F. N. Najm. Efficient rc power grid verification using node elimination. In *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pages 1–4, 2011.

[83] Luis Gravano, Gustavo D. Pifarré, Pablo E. Berman, and Jorge L. C. Sanz. Adaptive deadlock- and livelock-free routing with all minimal paths in torus networks. *IEEE Trans. Parallel Distrib. Syst.*, 5(12):1233–1251, December 1994. ISSN 1045-9219. doi: 10.1109/71.334898. URL http://dx.doi.org/10.1109/71.334898.

[84] C. Grecu, L. Anghel, P.P. Pande, A. Ivanov, and R. Saleh. Essential fault-tolerance metrics for NoC infrastructures. In *On-Line Testing Symposium IOLTS 07. 13th IEEE International*, pages 37 –42, july 2007. doi: 10.1109/IOLTS.2007.31.

[85] M. S. Gupta, J. L. Oatley, R. Joseph, Wei Gu-Yeon, and D. M. Brooks. Understanding voltage variations in chip multiprocessors using a distributed power-delivery network. In *Design, Automation and Test in Europe Conference and Exhibition, 2007. DATE '07*, pages 1–6, 2007.

[86] P.K. Hamedani, S. Hessabi, H. Sarbazi-Azad, and N.E. Jerger. Exploration of temperature constraints for thermal aware mapping of 3d networks on chip. In *Parallel, Distributed and Network-Based Processing (PDP), 2012 20th Euromicro International Conference on*, pages 499–506, 2012. doi: 10.1109/PDP.2012.68.

[87] Li Hang, J. Fan, Qi Zhenyu, S. X. D. Tan, Wu Lifeng, Y. Cai, and X. Hong. Partitioning-based approach to fast on-chip decoupling capacitor budgeting and minimization. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 25(11): 2402–2412, 2006.

[88] Wang Hang-Sheng, Zhu Xinping, Peh Li-Shiuan, and S. Malik. Orion: a power-performance simulator for interconnection net-

works. In *Microarchitecture. (MICRO-35). Proceedings. 35th Annual IEEE/ACM International Symposium on*, pages 294–305, 2002.

[89] Jim Held, Jerry Bautista, and Sean Koehl. White paper research at intel, from a few cores to many: A tera-scale computing research review, 2006. URL http://www.intel.com/content/www/us/en/research/intel-labs-tera-scale-research-paper.html.

[90] F. Hillier and G. Lieberman. *Introduction to Operations Research*. McGraw-Hill International Editions, 1995.

[91] R. Hoffmann, A. Prell, and T. Rauber. Dynamic task scheduling and load balancing on cell processors. In *Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on*, pages 205–212, 2010. doi: 10.1109/PDP.2010.24.

[92] Jingcao Hu and R. Marculescu. Energy-aware mapping for tile-based NoC architectures under performance constraints. In *Design Automation Conference, 2003. Proceedings of the ASP-DAC 2003. Asia and South Pacific*, pages 233–239, 2003.

[93] Jingcao Hu and R. Marculescu. Energy-aware communication and task scheduling for network-on-chip architectures under real-time constraints. In *Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings*, volume 1, pages 234–239 Vol.1, 2004. doi: 10.1109/DATE.2004.1268854.

[94] Jingcao Hu and R. Marculescu. Energy- and performance-aware mapping for regular NoC architectures. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 24(4):551 – 562, 2005. ISSN 0278-0070. doi: 10.1109/TCAD.2005.844106.

[95] Jingcao Hu and Radu Marculescu. DyAD - smart routing for networks-on-chip. In *In ACM/IEEE Design Automation Conference*, pages 260–263, 2004.

[96] Wei Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M.R. Stan. Hotspot: a compact thermal modeling methodology for early-stage vlsi design. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 14(5):501 –513, may 2006. ISSN 1063-8210. doi: 10.1109/TVLSI.2006.876103.

[97] Wei Huang, K. Sankaranarayanan, K. Skadron, R.J. Ribando, and M.R. Stan. Accurate, pre-rtl temperature-aware design using a parameterized, geometric thermal model. *Computers, IEEE Transactions on*, 57(9):1277 –1288, sept. 2008. ISSN 0018-9340. doi: 10.1109/TC.2008.64.

[98] Wei Huang, K. Skadron, S. Gurumurthi, R.J. Ribando, and M.R. Stan. Differentiating the roles of ir measurement and simulation

for power and temperature-aware design. In *Performance Analysis of Systems and Software, 2009. ISPASS 2009. IEEE International Symposium on*, pages 1 –10, april 2009. doi: 10.1109/ISPASS.2009.4919633.

[99] Wei Huang, M.R. Stan, S. Gurumurthi, R.J. Ribando, and K. Skadron. Interaction of scaling trends in processor architecture and cooling. In *Semiconductor Thermal Measurement and Management Symposium, 2010. SEMI-THERM 2010. 26th Annual IEEE*, pages 198–204, 2010. doi: 10.1109/STHERM.2010.5444290.

[100] W. Hung, C. Addo-Quaye, T. Theocharides, Y. Xie, N. Vijakrishnan, and M.J. Irwin. Thermal-aware ip virtualization and placement for networks-on-chip architecture. In *Computer Design: VLSI in Computers and Processors, 2004. ICCD 2004. Proceedings. IEEE International Conference on*, pages 430–437, 2004. doi: 10.1109/ICCD.2004.1347958.

[101] Intel. Many Integrated Core (MIC) Architecture. URL http://www.intel.com/content/www/us/en/architecture-and-technology/many-integrated-core/intel-many-integrated-core-architecture.html?wapkw=mic. [Nov. 26, 2013].

[102] ITRS. The International Technology Roadmap for Semiconductors (ITRS), 2004, http://www.itrs.net/, 2004.

[103] ITRS. The International Technology Roadmap for Semiconductors (ITRS), 2005, http://www.itrs.net/, 2005.

[104] ITRS. The International Technology Roadmap for Semiconductors (ITRS), Interconnects, 2011, http://www.itrs.net/, 2011.

[105] L. Jain, B. Al-Hashimi, M Zwolinski, M. Gaur, P. Rosinger, and V. Laxmi. NIRGAM: a simulator for NoC interconnect routing and application modeling. URL http://nirgam.ecs.soton.ac.uk/. [Dec 02, 2013].

[106] Antoine Jalabert, Srinivasan Murali, Luca Benini, and Giovanni De Micheli. × pipescompiler: A tool for instantiating application specific networks on chip. In *Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings*, volume 2, pages 884–889. IEEE, 2004.

[107] Wooyoung Jang and David Pan. A3map: architecture-aware analytic mapping for networks-on-chip. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 17(3):26, 2012.

[108] Jedec. Failure mechanisms and models for semiconductor devices. *JEDEC Publication JEP122-A*, 2002.

[109] Natalie Enright Jerger and Li-Shiuan Peh. On-chip networks. *Synthesis Lectures on Computer Architecture*, 4(1):1–141, 2009.

[110] Xu Jiang, Wolf Wayne, Henkel Joerg, and Chakradhar Srimat. A design methodology for application-specific networks-on-chip. *ACM Trans. Embed. Comput. Syst.*, 5(2):263–280, 2006. 1151076.

[111] N. Kozhaya Joseph. Fast power grid simulation. In R. Nassif Sani, editor, *37th Conference on Design Automation (DAC'00)*, volume 0, pages 156–161, 2000.

[112] A. B. Kahng, B. Li, L. S. Peh, and K. Samadi. Orion 2.0: A power-area simulator for interconnection networks. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, PP(99):1–5, 2011.

[113] Ammar Jallawi Karkar, Janice E. Turner, Kenneth Tong, Ra'ed AI-Dujaily, Terrence Mak, Alex Yakovlev, and Fei Xia. Hybrid wire-surface wave interconnects for next-generation networks-on-chip. *Institution of Engineering and Technology*, pages – (0), 2013. URL http://digital-library.theiet.org/content/journals/10.1049/iet-cdt.2013.0030.

[114] M.M. Khan, D.R. Lester, L.A. Plana, A. Rast, X. Jin, E. Painkras, and S.B. Furber. Spinnaker: Mapping neural networks onto a massively-parallel chip multiprocessor. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 2849–2856, 2008. doi: 10.1109/IJCNN.2008.4634199.

[115] N.H. Khan, S.M. Alam, and S. Hassoun. Power delivery design for 3-d ics using different through-silicon via (tsv) technologies. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 19(4):647 –658, april 2011. ISSN 1063-8210. doi: 10.1109/TVLSI.2009.2038165.

[116] S. Kirolos, Y. Massoud, and Y. Ismail. Accurate analytical delay modeling of cmos clock buffers considering power supply variations. In *Circuits and SystemsISCAS. IEEE International Symposium on*, pages 3394–3397, 2008.

[117] S. Kirolos, Y. Massoud, and Y. Ismail. Power-supply-variation-aware timing analysis of synchronous systems. In *Circuits and Systems ISCAS. IEEE International Symposium on*, pages 2418–2421, 2008.

[118] J. N. Kozhaya, S. R. Nassif, and F. N. Najm. A multigrid-like technique for power grid analysis. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 21(10):1148–1160, 2002.

[119] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani. A network on chip architecture and design methodology. In *VLSI, 2002. Proceedings. IEEE Computer Society Annual Symposium on*, pages 105–112, 2002. doi: 10.1109/ISVLSI.2002.1016885.

[120] S. Kvatinsky, E.G. Friedman, A. Kolodny, and L. Scha? andchter. Power grid analysis based on a macro circuit model. In *Electrical and Electronics Engineers in Israel (IEEEI), 2010 IEEE 26th Convention of*, pages 000708 –000712, nov. 2010. doi: 10.1109/EEEI.2010.5662121.

[121] A. Lankes, T. Wild, A. Herkersdorf, S. Sonntag, and H. Reinig. Comparison of deadlock recovery and avoidance mechanisms to approach message dependent deadlocks in on-chip networks. In *Networks-on-Chip (NOCS), 2010 Fourth ACM/IEEE International Symposium on*, pages 17–24, 2010. doi: 10.1109/NOCS.2010.11.

[122] Shu-Yen Lin, Tzu-Chu Yin, Hao-Yu Wang, and An-Yeu Wu. Traffic-and thermal-aware routing for throttled three-dimensional network-on-chip systems. In *VLSI Design, Automation and Test (VLSI-DAT), 2011 International Symposium on*, pages 1 –4, april 2011. doi: 10.1109/VDAT.2011.5783639.

[123] Jian Liu, L-R Zheng, Dinesh Pamunuwa, and Hannu Tenhunen. A global wire planning scheme for network-on-chip. In *Circuits and Systems, 2003. ISCAS'03. Proceedings of the 2003 International Symposium on*, volume 4, pages IV–892. IEEE, 2003.

[124] James Jian-Qiang Lu. 3D integration: Why, what, who, when? In *Future Fab Intl.*, volume Issue 23, 9 2007.

[125] Zhonghai Lu, Lei Xia, and A. Jantsch. Cluster-based simulated annealing for mapping cores onto 2d mesh networks on chip. In *Design and Diagnostics of Electronic Circuits and Systems, 2008. DDECS 2008. 11th IEEE Workshop on*, pages 1–6, 2008. doi: 10.1109/DDECS.2008.4538763.

[126] Chiao-Ling Lung, Yi-Lun Ho, Ding-Ming Kwai, and Shih-Chieh Chang. Thermal-aware on-line task allocation for 3D multi-core processor throughput optimization. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2011*, pages 1 –6, march 2011.

[127] Nir Magen, Avinoam Kolodny, Uri Weiser, and Nachum Shamir. Interconnect-power dissipation in a microprocessor. In *Proceedings of the 2004 international workshop on System level interconnect prediction*, SLIP '04, pages 7–13, New York, NY, USA, 2004. ACM. ISBN 1-58113-818-0. doi: 10.1145/966747.966750. URL http://doi.acm.org/10.1145/966747.966750.

[128] T. Mak, P. Y. K. Cheung, K.-P. Lam, and W. Luk. Adaptive routing in network-on-chips using a dynamic-programming network. *Industrial Electronics, IEEE Transactions on*, 58(8):3701 –3716, aug. 2011. ISSN 0278-0046. doi: 10.1109/TIE.2010.2081953.

[129] Terrence Mak, Peter Y.K. Cheung, Wayne Luk, and Kai Pui Lam. A DP-network for optimal dynamic routing in network-on-chip. In *CODES+ISSS '09*, pages 119–128. ACM, 2009. ISBN 978-1-60558-628-1. doi: http://doi.acm.org/10.1145/1629435.1629452.

[130] R. Marculescu, U.Y. Ogras, Li-Shiuan Peh, N.E. Jerger, and Y. Hoskote. Outstanding research problems in noc design: System, microarchitecture, and circuit perspectives. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 28 (1):3–21, 2009. ISSN 0278-0070. doi: 10.1109/TCAD.2008.2010691.

[131] Radu Marculescu and Paul Bogdan. The chip is the network: Toward a science of network-on-chip design. *Foundations and trends in electronic design automation*, 2(4):371–461, 2009.

[132] Mikael Millberg, Erland Nilsson, Rikard Thid, and Axel Jantsch. Guaranteed bandwidth using looped containers in temporally disjoint networks within the nostrum network on chip. In *Proceedings of the Conference on Design, Automation and Test in Europe - Volume 2*, DATE '04, pages 20890–, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2085-5. URL http://dl.acm.org/citation.cfm?id=968879.969206.

[133] D. Miller. Device requirements for optical interconnects to silicon chips. *Proceedings of the IEEE*, 97(7):1166 –1185, july 2009. ISSN 0018-9219. doi: 10.1109/JPROC.2009.2014298.

[134] F. Mohamood, M. B. Healy, Lim Sung Kyu, and H. H. S. Lee. Noise-direct: A technique for power supply noise aware floorplanning using microarchitecture profiling. In *Design Automation Conference. ASP-DAC '07. Asia and South Pacific*, pages 786–791, 2007.

[135] R. Mullins. Netmaker: Interconnection Network Simulator. URL http://www-dyn.cl.cam.ac.uk/~rdm34/wiki/index.php?title=Main_Page. [Dec 02, 2013].

[136] S. Murali, L. Benini, and G. De Micheli. Mapping and physical planning of networks-on-chip architectures with quality-of-service guarantees. In *Design Automation Conference, 2005. Proceedings of the ASP-DAC 2005. Asia and South Pacific*, volume 1, pages 27–32 Vol. 1, 2005. doi: 10.1109/ASPDAC.2005.1466124.

[137] Srinivasan Murali and Giovanni De Micheli. Bandwidth-constrained mapping of cores onto NoC architectures. In *Proceedings of the conference on Design, automation and test in Europe*

- *Volume 2*, DATE '04, pages 896 – 901 Vol.2, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2085-5.

[138] Srinivasan Murali, Paolo Meloni, Federico Angiolini, David Atienza, Salvatore Carta, Luca Benini, Giovanni De Micheli, and Luigi Raffo. Designing application-specific networks on chips with floorplan information. In *Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, pages 355–362. ACM, 2006.

[139] F.N. Najm. Transition density: a new measure of activity in digital circuits. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 12(2):310 –323, feb 1993. ISSN 0278-0070. doi: 10.1109/43.205010.

[140] Dr. Sani R. Nassif. IBM power grid benchmarks, 2008.

[141] S. R. Nassif. Power grid analysis benchmarks. In *Design Automation Conference, 2008. ASPDAC 2008. Asia and South Pacific*, pages 376–381, 2008.

[142] L.M. Ni and P.K. McKinley. A survey of wormhole routing techniques in direct networks. *Computer*, 26(2):62 –76, feb 1993. ISSN 0018-9162. doi: 10.1109/2.191995.

[143] S.K. Nithin, G. Shanmugam, and S. Chandrasekar. Dynamic voltage (ir) drop analysis and design closure: Issues and challenges. In *Quality Electronic Design (ISQED), 2010 11th International Symposium on*, pages 611 –617, march 2010. doi: 10.1109/ISQED.2010.5450515.

[144] N. Nordbotten, M. Gomez, J. Flich, P. Lopez, A. Robles, T. Skeie, O. Lysne, and J. Duato. A fully adaptive fault-tolerant routing methodology based on intermediate nodes. *Network and Parallel Computing*, pages 341–356, 2004.

[145] Umit Y. Ogras, Jingcao Hu, and Radu Marculescu. Key research problems in noc design: a holistic perspective. In *Hardware/Software Codesign and System Synthesis, 2005. CODES+ISSS '05. Third IEEE/ACM/IFIP International Conference on*, pages 69 –74, sept. 2005. doi: 10.1145/1084834.1084856.

[146] Luciano Ost, Marcelo Mandelli, Gabriel Marchesan Almeida, Leandro Moller, Leandro Soares Indrusiak, Gilles Sassatelli, Pascal Benoit, Manfred Glesner, Michel Robert, and Fernando Moraes. Power-aware dynamic mapping heuristics for noc-based mpsocs using a unified model-based approach. *ACM Trans. Embed. Comput. Syst.*, 12(3):75:1–75:22, April 2013. ISSN 1539-9087. doi: http://dx.doi.org/10.1145/2442116.2442125. URL http://doi.acm.org/http://dx.doi.org/10.1145/2442116.2442125.

[147] J.D. Owens, W.J. Dally, R. Ho, D. N. Jayasimha, S.W. Keckler, and Li-Shiuan Peh. Research challenges for on-chip interconnection networks. *Micro, IEEE*, 27(5):96–108, 2007. ISSN 0272-1732. doi: 10.1109/MM.2007.4378787.

[148] E. Painkras, L.A. Plana, J. Garside, S. Temple, S. Davidson, J. Pepper, D. Clark, C. Patterson, and S. Furber. Spinnaker: A multi-core system-on-chip for massively-parallel neural net simulation. In *Custom Integrated Circuits Conference (CICC), 2012 IEEE*, pages 1–4, 2012. doi: 10.1109/CICC.2012.6330636.

[149] M. Palesi, R. Holsmark, S. Kumar, and V. Catania. Application specific routing algorithms for networks on chip. *Parallel and Distributed Systems, IEEE Transactions on*, 20(3):316–330, 2009. ISSN 1045-9219. doi: 10.1109/TPDS.2008.106.

[150] Maurizio Palesi and Masoud Daneshtalab. *Routing Algorithms in Networks-on-Chip*. Springer, 2014. ISBN 978-1-4614-8274-1.

[151] P.P. Pande, C. Grecu, A. Ivanov, and R. Saleh. High-throughput switch-based interconnect for future socs. In *System-on-Chip for Real-Time Applications, 2003. Proceedings. The 3rd IEEE International Workshop on*, pages 304 – 310, june-2 july 2003. doi: 10.1109/IWSOC.2003.1213053.

[152] Michael K. Papamichael and James C. Hoe. Connect: Re-examining conventional wisdom for designing nocs in the context of fpgas. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, FPGA '12, pages 37–46, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1155-7. doi: 10.1145/2145694.2145703. URL http://doi.acm.org/10.1145/2145694.2145703.

[153] Sudeep Pasricha. Exploring serial vertical interconnects for 3d ics. In *Proceedings of the 46th Annual Design Automation Conference*, DAC '09, pages 581–586, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-497-3. doi: 10.1145/1629911.1630061. URL http://doi.acm.org/10.1145/1629911.1630061.

[154] M. Pedram and S. Nazarian. Thermal modeling, analysis, and management in vlsi circuits: Principles and methods. *Proceedings of the IEEE*, 94(8):1487–1501, 2006. ISSN 0018-9219. doi: 10.1109/JPROC.2006.879797.

[155] Chang Po-Hao and Chen Jia-Ming. A decoupling technique on switch factor based analysis of rlc interconnects. In *Electro/Information Technology IEEE International Conference on*, pages 73–78, 2007.

[156] Mikhhail Popovich, Andrey V Mezhiba, and Eby G Friedman. *Power distribution networks with on-chip decoupling capacitors*. Springer, 2008. ISBN 978-0-387-71601-5.

[157] Zhiliang Qian and Chi-Ying Tsui. A thermal-aware application specific routing algorithm for network-on-chip design. In *Design Automation Conference (ASP-DAC), 2011 16th Asia and South Pacific*, pages 449–454, 2011. doi: 10.1109/ASPDAC.2011.5722232.

[158] Jan M. Rabaey, Anantha P. Chandrakasan, and Borivoje Nikolic. *Digital integrated circuits : a design perspective*. Prentice Hall electronics and VLSI series. Pearson Education, January 2003. ISBN 0130909963.

[159] M.S. Rahaman and M.H. Chowdhury. Crosstalk avoidance and error-correction coding for coupled rlc interconnects. In *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pages 141–144, 2009. doi: 10.1109/ISCAS.2009.5117705.

[160] C. Rusu, L. Anghel, and D. Avresky. Message routing in 3D networks-on-chip. In *NORCHIP, 2009*, pages 1 –4, nov. 2009. doi: 10.1109/NORCHP.2009.5397855.

[161] M. Sadri, A. Bartolini, and L. Benini. Single-chip cloud computer thermal model. In *Thermal Investigations of ICs and Systems (THERMINIC), 2011 17th International Workshop on*, pages 1–6, 2011.

[162] Pradip Kumar Sahu and Santanu Chattopadhyay. A survey on application mapping strategies for network-on-chip design. *Journal of Systems Architecture*, 59(1):60 – 76, 2013. ISSN 1383-7621. doi: http://dx.doi.org/10.1016/j.sysarc.2012.10.004. URL http://www.sciencedirect.com/science/article/pii/S1383762112000902.

[163] M. Saint-Laurent and M. Swaminathan. Impact of power-supply noise on timing in high-frequency microprocessors. *IEEE Transactions on Advanced Packaging*, 27(1):135–144, 2004.

[164] Praveen Salihundam, Shailendra Jain, Tiju Jacob, Shasi Kumar, Vasantha Erraguntla, Yatin Hoskote, Sriram Vangal, Gregory Ruhl, and Nitin Borkar. A 2 tb/s 6× 4 mesh network for a single-chip cloud computer with dvfs in 45 nm cmos. *IEEE journal of solid-state circuits*, 46(4):757–766, 2011.

[165] The IntelÂ® Tera scale Computing Research Program. Intel: Single-chip cloud computer. URL http://techresearch.intel.com/ProjectDetails.aspx?Id=1. [Nov. 26, 2013].

[166] L.W. Schaper, S.L. Burkett, S. Spiesshoefer, G.V. Vangara, Z. Rahman, and S. Polamreddy. Architectural implications and process

development of 3-d vlsi -axis interconnects using through silicon vias. *Advanced Packaging, IEEE Transactions on*, 28(3):356 – 366, aug. 2005. ISSN 1521-3323. doi: 10.1109/TADVP.2005.853271.

[167] C. Seiculescu, S. Murali, L. Benini, and G. De Micheli. Sunfloor 3d: A tool for networks on chip topology synthesis for 3-d systems on chips. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 29(12):1987–2000, 2010.

[168] O. Semenov, A. Vassighi, and M. Sachdev. Impact of self-heating effect on long-term reliability and performance degradation in cmos circuits. *Device and Materials Reliability, IEEE Transactions on*, 6(1):17 – 27, march 2006. ISSN 1530-4388. doi: 10.1109/TDMR.2006.870340.

[169] L. Shang, Li-Shiuan Peh, A Kumar, and N.K. Jha. Temperature-aware on-chip networks. *Micro, IEEE*, 26(1):130–139, 2006. ISSN 0272-1732. doi: 10.1109/MM.2006.23.

[170] Li Shang, L. Peh, A. Kumar, and N.K. Jha. Thermal modeling, characterization and management of on-chip networks. In *Microarchitecture, 2004. MICRO-37 2004. 37th International Symposium on*, pages 67 – 78, dec. 2004. doi: 10.1109/MICRO.2004.35.

[171] J. M. S. Silva, J. R. Phillips, and L. M. Silveira. Efficient representation and analysis of power grids. In *Design, Automation and Test in Europe*, pages 420–425, 2008.

[172] K. Skadron, M.R. Stan, W. Huang, Sivakumar Velusamy, Karthik Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. In *Computer Arch., 2003. Proc. 30th Annual International Symp. on*, pages 2 – 13, june 2003. doi: 10.1109/ISCA.2003.1206984.

[173] P. P. Sotiriadis and A. P. Chandrakasan. A bus energy model for deep submicron technology. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 10(3):341–350, 2002.

[174] J. Srinivasan, S.V. Adve, P. Bose, and J.A. Rivers. The case for lifetime reliability-aware microprocessors. pages 276 – 287, june 2004. doi: 10.1109/ISCA.2004.1310781.

[175] J. Srinivasan, S.V. Adve, P. Bose, and J.A. Rivers. The impact of technology scaling on lifetime reliability. In *Dependable Systems and Networks, 2004 International Conference on*, pages 177 – 186, june-1 july 2004. doi: 10.1109/DSN.2004.1311888.

[176] Frits Steenhof, Harry Duque, Björn Nilsson, Kees Goossens, and Rafael Peset Llopis. Networks on chips for high-end consumer-electronics tv system architectures. In *Proceedings of the conference*

on Design, automation and test in Europe: Designers' forum, DATE '06, pages 148–153, 3001 Leuven, Belgium, Belgium, 2006. European Design and Automation Association. ISBN 3-9810801-0-6. URL http://dl.acm.org/citation.cfm?id=1131355.1131387.

[177] R.R. Tamhankar, S. Murali, and G. De Micheli. Performance driven reliable link design for networks on chips. In *Design Automation Conference, 2005. Proceedings of the ASP-DAC 2005. Asia and South Pacific*, volume 2, pages 749–754 Vol. 2, 2005. doi: 10.1109/ASPDAC.2005.1466449.

[178] N. Tanaka, Y. Yoshimura, M. Kawashita, T. Uematsu, C. Miyazaki, N. Toma, K. Hanada, M. Nakanishi, T. Naito, T. Kikuchi, and T. Akazawa. Through-silicon via interconnection for 3D integration using room-temperature bonding. *Advanced Packaging, IEEE Transactions on*, 32(4):746 –753, nov. 2009. ISSN 1521-3323. doi: 10.1109/TADVP.2009.2027420.

[179] Ahonen Tapani, A. Sig David, Tortosa enza, Bin Hong, and Nurmi Jari. Topology optimization for application-specific networks-on-chip, 2004. 966758 53-60.

[180] G. Tarawneh, T. Mak, and A. Yakovlev. Intra-chip physical parameter sensor for fpgas using flip-flop metastability. In *Field Programmable Logic and Applications (FPL), 2012 22nd International Conference on*, pages 373–379, 2012. doi: 10.1109/FPL.2012.6339207.

[181] M.B. Taylor, J. Kim, J. Miller, D. Wentzlaff, F. Ghodrat, B. Greenwald, H. Hoffman, P. Johnson, Jae-Wook Lee, W. Lee, A. Ma, A. Saraf, M. Seneski, N. Shnidman, V. Strumpen, M. Frank, S. Amarasinghe, and A. Agarwal. The raw microprocessor: a computational fabric for software circuits and general-purpose programs. *Micro, IEEE*, 22(2):25–35, 2002. ISSN 0272-1732. doi: 10.1109/MM.2002.997877.

[182] Michael Bedford Taylor, Walter Lee, Jason Miller, David Wentzlaff, Ian Bratt, Ben Greenwald, Henry Hoffmann, Paul Johnson, Jason Kim, James Psota, et al. Evaluation of the raw microprocessor: An exposed-wire-delay architecture for ilp and streams. In *ACM SIGARCH Computer Architecture News*, volume 32, page 2. IEEE Computer Society, 2004.

[183] Tilera. Tilepro processor family, Dec 2011. URL http://www.tilera.com/products/processors/TILEPro_Family.

[184] A. Todri and M. Marek-Sadowska. Power delivery for multi-core systems. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 19(12):2243–2255, 2011. ISSN 1063-8210. doi: 10.1109/TVLSI.2010.2080694.

[185] A. Todri, M. Marek-Sadowska, and J. Kozhaya. Power supply noise aware workload assignment for multi-core systems. In *Computer-Aided Design, 2008. ICCAD 2008. IEEE/ACM International Conference on*, pages 330–337, 2008.

[186] R. Tornero, V. Sterrantino, M. Palesi, and J.M. Orduna. A multi-objective strategy for concurrent mapping and routing in networks on chip. In *Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–8, 2009. doi: 10.1109/IPDPS.2009.5161128.

[187] Suleyman Tosun. Cluster-based application mapping method for network-on-chip. *Advances in Engineering Software*, 42(10):868 – 874, 2011. ISSN 0965-9978. doi: http://dx.doi.org/10.1016/j.advengsoft.2011.06.005. URL http://www.sciencedirect.com/science/article/pii/S0965997811001669.

[188] Anh T. Tran and Bevan Baas. Noctweak: a highly parameterizable simulator for early exploration of performance and energy of networks on-chip. Technical Report ECE-VCL-2012-2, VLSI Computation Lab, ECE Department, University of California, Davis, 2012. http://www.ece.ucdavis.edu/vcl/pubs/2012.07.techreport.noctweak/.

[189] S. Tuuna, L. R. Zheng, J. Isoaho, and H. Tenhunen. Modeling of on-chip bus switching current and its impact on noise in power supply grid. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 16(6):766–770, 2008.

[190] A.S. Vaidya, A. Sivasubramaniam, and C.R. Das. Impact of virtual channels and adaptive routing on application performance. *Parallel and Distributed Systems, IEEE Transactions on*, 12(2):223 –237, feb 2001. ISSN 1045-9219. doi: 10.1109/71.910875.

[191] Leslie G Valiant and Gordon J Brebner. Universal schemes for parallel communication. In *Proceedings of the thirteenth annual ACM symposium on Theory of computing*, pages 263–277. ACM, 1981.

[192] S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, P. Iyer, A. Singh, and T. Jacob. An 80-tile 1.28 tflops network-on-chip in 65nm cmos. In *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, pages 98–589. IEEE, 2007. Solid-State Circuits Conference. ISSCC '07. Digest of Technical Papers. IEEE International.

[193] S. R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar. An 80-tile sub-100-w

teraflops processor in 65-nm cmos. *IEEE Journal of Solid-State Circuits*, 43(1):29–41, 2008.

[194] Praveen Vellanki, Nilanjan Banerjee, and Karam S. Chatha. Quality-of-service and error control techniques for mesh-based network-on-chip architectures. *Integr. VLSI J.*, 38(3):353–382, January 2005. ISSN 0167-9260. doi: 10.1016/j.vlsi.2004.07.009. URL http://dx.doi.org/10.1016/j.vlsi.2004.07.009.

[195] S. Velusamy, Wei Huang, J. Lach, M. Stan, and K. Skadron. Monitoring temperature in fpga based socs. In *Computer Design: VLSI in Computers and Processors, 2005. ICCD 2005. Proceedings. 2005 IEEE International Conference on*, pages 634–637, 2005. doi: 10.1109/ICCD.2005.78.

[196] H. Wang, L. S. Peh, and S. Malik. Power-driven design of router microarchitectures in on-chip networks. In *Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*, page 105. IEEE Computer Society, 2003.

[197] Y. Wang, J. Xu, Y. Xu, W. Liu, and H. Yang. Power gating aware task scheduling in mpsoc. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, (99):1–12, 2010.

[198] Roshan Weerasekera, Dinesh Pamunuwa, Li-Rong Zheng, and Hannu Tenhunen. Two-dimensional and three-dimensional integration of heterogeneous electronic systems under cost, performance, and technological constraints. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 28(8): 1237–1250, 2009.

[199] Awet Yemane Weldezion, Matt Grange, Dinesh Pamunuwa, Zhonghai Lu, Axel Jantsch, Roshan Weerasekera, and Hannu Tenhunen. Scalability of network-on-chip communication architecture for 3-d meshes. In *Proceedings of the 2009 3rd ACM/IEEE International Symposium on Networks-on-Chip*, pages 114–123. IEEE Computer Society, 2009.

[200] A.Y. Weldezion, M. Grange, D. Pamunuwa, Zhonghai Lu, A. Jantsch, R. Weerasekera, and H. Tenhunen. Scalability of network-on-chip communication architecture for 3-d meshes. In *Networks-on-Chip, 2009. NoCS 2009. 3rd ACM/IEEE International Symposium on*, pages 114–123, 2009. doi: 10.1109/NOCS.2009.5071459.

[201] P. Wilkerson, A. Raman, and M. Turowski. Fast, automated thermal simulation of three-dimensional integrated circuits. In *Thermal and Thermomechanical Phenomena in Electronic Systems, 2004. ITHERM '04. The Ninth Intersociety Conference on*, pages 706 – 713 Vol.1, june 2004. doi: 10.1109/ITHERM.2004.1319245.

[202] Jie Wu. A simple fault-tolerant adaptive and minimal routing approach in 3-d meshes. *J. Comput. Sci. Technol.*, 18(1):1–13, January 2003. ISSN 1000-9000. doi: 10.1007/BF02946645. URL http://dx.doi.org/10.1007/BF02946645.

[203] Yuan Xie and W. Wolf. Allocation and scheduling of conditional task graph in hardware/software co-synthesis. In *Design, Automation and Test in Europe, 2001. Conference and Exhibition 2001. Proceedings*, pages 620–625, 2001. doi: 10.1109/DATE.2001.915088.

[204] Licheng Xue, Yujin Gao, and Jibin Fu. A high performance 3D interconnection network for many-core processors. In *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, volume 1, pages V1–383 –V1–389, april 2010. doi: 10.1109/ICCET.2010.5486092.

[205] Shan Yan and Bill Lin. Design of application-specific 3d networks-on-chip architectures. In *Computer Design, 2008. ICCD 2008. IEEE International Conference on*, pages 142–149, 2008. doi: 10.1109/ICCD.2008.4751853.

[206] T.T. Ye, L. Benini, and G. De Micheli. Analysis of power consumption on switch fabrics in network routers. In *Design Automation Conference, 2002. Proceedings. 39th*, pages 524 – 529, 2002. doi: 10.1109/DAC.2002.1012681.

[207] M. Zhao, R. V. Panda, S. S. Sapatnekar, and D. Blaauw. Hierarchical analysis of power distribution networks. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 21 (2):159–168, 2002.

[208] L. R. Zheng and H. Tenhunen. Fast modeling of core switching noise on distributed lrc power grid in ulsi circuits. In *Electrical Performance of Electronic Packaging, 2000, IEEE Conference on.*, pages 307–310, 2000.

[209] Changyun Zhu, Zhenyu Gu, Li Shang, R.P. Dick, and R. Joseph. Three-dimensional chip-multiprocessor run-time thermal management. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 27(8):1479 –1492, aug. 2008. ISSN 0278-0070. doi: 10.1109/TCAD.2008.925793.

[210] Kenneth M. Zick and John P. Hayes. On-line sensing for healthier fpga systems. In *Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays*, FPGA '10, pages 239–248, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-911-4. doi: 10.1145/1723112.1723153. URL http://doi.acm.org/10.1145/1723112.1723153.

[211] H. Zimmer and A. Jantsch. A fault model notation and error-control scheme for switch-to-switch buses in a network-on-chip. In *Hardware/Software Codesign and System Synthesis, 2003. First IEEE/ACM/IFIP International Conference on*, pages 188 –193, oct. 2003. doi: 10.1109/CODESS.2003.1275281.