

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DE COMPUTAÇÃO**

**3D NETWORK-ON-CHIP
ARCHITECTURAL EXPLORATION**

YAN GHIDINI DE SOUZA

Advisor: César Augusto Missio Marcon
Co-Advisor: Thais Christina Webber dos Santos

Porto Alegre, Brazil
January 2014

Dados Internacionais de Catalogação na Publicação (CIP)

S729t Souza, Yan Ghidini de
3D Network-On- Chip architectural exploration / Yan Ghidini
de Souza. – Porto Alegre, 2014.
88 p.

Diss. (Mestrado) – Fac. de Informática, PUCRS.
Orientador: Prof. Dr. César Augusto Missio Marcon.
Co-orientador: Thaís Christina Webber dos Santos.

1. Informática. 2. Arquitetura de Computador.
3. Multiprocessadores. I. Marcon, César Augusto Missio. II.
Santos, Thaís Christina Webber dos. III. Título.

CDD 004.35

**Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS**



Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

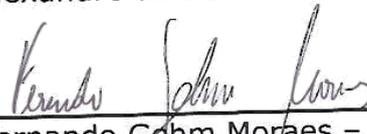
TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

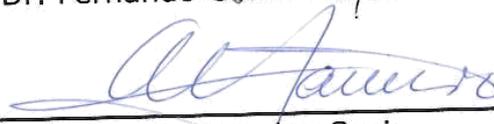
Dissertação intitulada "3D Network on Chip Architectural Exploration" apresentada por Yan Ghidini de Souza como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, aprovada em 12/03/2014 pela Comissão Examinadora:


Prof. Dr. César Augusto Missio Marcon - PPGCC/PUCRS
Orientador


Dra. Thais Christina Webber dos Santos - PPGEE/PUCRS
Coorientadora


Prof. Dr. Alexandre de Moraes Amory - PPGCC/PUCRS


Prof. Dr. Fernando Gehm Moraes - PPGCC/PUCRS


Prof. Dr. Altamiro Amadeu Susin - UFRGS

Homologada em 24/04/2014, conforme Ata No. 006 pela Comissão Coordenadora.


Prof. Dr. Luiz Gustavo Leão Fernandes
Coordenador.

PUCRS

Campus Central
Av. Ipiranga, 6681 - P32- sala 507 - CEP: 90619-900
Fone: (51) 3320-3611 - Fax (51) 3320-3621
E-mail: ppgcc@pucrs.br
www.pucrs.br/facin/pos

AGRADECIMENTOS

A conclusão desta dissertação traz consigo uma ponderação e uma reflexão sobre esta longa e gratificante trajetória, desde a minha decisão em realizar este curso com todas as responsabilidades que ele impunha. Ao final deste caminho percorrido, acredito ser impossível deixar de agradecer às pessoas que, de diferentes formas, contribuíram para a realização de mais esta etapa em minha vida.

Agradeço profundamente ao meu orientador, Prof. Dr. César Marcon, pela oportunidade e por ter dedicado a mim sua confiança e ensinamentos desde 2007, ano em que fui bolsista de iniciação científica. Pelo comprometimento e ética demonstrados durante todos esses anos.

A minha coorientadora, Prof.^a Dra. Thais Webber, pelo apoio, confiança e por todas as contribuições e revisões durante a realização deste trabalho.

Ao Prof. Msc. Matheus Trevisan, por toda a ajuda, novas ideias, e excelentes e imprescindíveis colaborações nos artigos publicados.

Aos bolsistas de iniciação científica Fernando Grando e Lucas Brahm, por todo o auxílio e esforço direcionado a este trabalho.

De forma geral, a todos os professores do PPGCC que contribuíram para o meu crescimento profissional e pessoal, dentre eles, Alexandre Amory, Ney Calazans, Fernando Moraes e Fabiano Hessel. Um agradecimento especial ao Prof. Dr. Edson Moreno, que colaborou muito com meu aprendizado na área de redes intrachip durante meu trabalho de conclusão de curso e também no período inicial do mestrado.

Aos amigos e colegas, pela convivência, momentos de descontração e também muito trabalho juntos.

A todos os meus familiares, em especial a minha mãe Marilene, que sempre acreditou em mim e suportou minhas decisões incondicionalmente. Ao meu pai Wilson, que sempre confiou nas minhas capacidades. Ao meu irmão Roges, que sempre me incentivou a realizar este curso.

A minha esposa Laís, que me inspira por sua sabedoria e pragmatismo. Este objetivo não seria completo sem o teu apoio, carinho, compreensão e amor (Bolla, eu te amo!).

3D NETWORK-ON-CHIP ARCHITECTURAL EXPLORATION

RESUMO

Comunicação desempenha papel fundamental em projetos de Sistemas Multiprocessados em Chips (MPSoCs, do inglês *Multiprocessor Systems-on-Chips*). Desta maneira, Redes Intrachip (NoCs, do inglês *Networks-on-Chips*) têm sido propostas como solução para a comunicação global em MPSoCs complexos. Arquiteturas baseadas em NoCs são caracterizadas por vários compromissos relacionados a características estruturais, a especificações de desempenho e a demandas da aplicação. Adicionalmente, o atraso na comunicação e a dissipação de potência estão aumentando conforme o número de núcleos em uma camada 2D (bidimensional) aumenta. Uma das razões para isso é o longo diâmetro da rede e a distância de comunicação entre núcleos. Neste cenário, a tecnologia de Circuito Integrado (CI) 3D (tridimensional) aplicada às arquiteturas do tipo NoC permite maior integração entre dispositivos e com interconexões menores, e possibilita também reduzir o tamanho e o número de interconexões globais (conexões entre todos os elementos de uma rede), o que, por sua vez, influencia diretamente o desempenho da comunicação e permite oportunidades para inovações em arquiteturas de chips. Ademais, arquiteturas baseadas em NoCs 3D aparecem como alternativa à redução de indicadores como latência, consumo de energia e área quando comparadas às topologias de NoCs 2D. Embora existam diversas tecnologias disponíveis para interconexões em redes 3D, a utilização de *Through Silicon Vias* (TSVs) é uma abordagem viável como interconexão entre camadas empilhadas. Entretanto, a desvantagem que a TSV ocasiona nas atuais tecnologias 3D é que tais interconexões são geralmente custosas em termos de área de silício, o que acarreta limitações no seu uso.

Este trabalho apresenta uma arquitetura de NoC 3D do tipo malha chamada Lasio, explorando impactos arquiteturais e comparando duas topologias, uma 3D e outra 2D, em termos de latência, vazão e ocupação de buffers. O presente trabalho também analisa a influência da profundidade dos buffers de entrada das portas dos roteadores nas latências de comunicação e de aplicação. Tais avaliações consideraram diferentes parâmetros de rede, como por exemplo, padrões de tráfego, profundidade dos buffers, nível de serialização das TSVs e uma variedade de tamanhos de pacotes. Além disso, durante este trabalho, foi implementado um esquema de serialização de TSV na Lasio. Em seguida, foi analisado o impacto de diferentes níveis de serialização no custo de área, na

dissipação de potência, nas latências de rede e de aplicação e na ocupação dos buffers de entrada das portas de cada roteador em uma NoC 3D 4x4x4 do tipo malha.

Dentre os resultados alcançados durante este trabalho, foi verificado que topologias 3D quando comparadas a topologias 2D minimizam em 30% a latência de aplicação e aumentam 56% a vazão dos pacotes. Além disso, este trabalho salienta que quando é aplicado um tamanho de buffer apropriado, a latência de aplicação é reduzida até 3,4 vezes para topologias 2D e 2,3 vezes para topologias 3D. Resultados adicionais demonstram que NoCs 3D reduzem mais a ocupação das conexões internas quando comparadas com NoCs equivalentes 2D, o que potencialmente permite maior vazão e maior eficiência com relação à dissipação de potência e latência. Ademais, os resultados também demonstraram que o esquema de serialização proposto permite reduzir o uso de TSVs com uma baixa perda de desempenho, o que ressalta potenciais benefícios do esquema em MPSoCs baseados em NoCs 3D.

Palavras Chave: MPSoC, NoC 3D, Circuito Integrado 3D, Through Silicon Via (TSV), Serialização de TSVs.

3D NETWORK-ON-CHIP ARCHITECTURAL EXPLORATION

ABSTRACT

Communication plays a crucial role in high performance design of Multiprocessor Systems-on-Chips (MPSoCs). Accordingly, Networks-on-Chip (NoCs) have been proposed as a solution to deal with the global communication of complex MPSoCs. NoC-based architectures are characterized by various tradeoffs related to structural characteristics, performance specifications, and application demands. Additionally, wire delay and power dissipation are rising as the number of cores over a 2D (two-dimensional) plane increases. One of the reasons for that is the long network diameter and overall communication distance. In this scenario, 3D (three-dimensional) Integrated Circuit (IC) technology applied to NoC architectures allows greater device integration, shorter interconnection, and it aims to reduce the length and number of global interconnections (interconnections among every processing element), which directly influences on the communication performance and allows opportunities for chip architecture innovations. Moreover, 3D NoC-based architectures appear as alternative to reduce network latency, energy consumption and area footprint in comparison to 2D NoC topologies. Albeit a wide variety of technologies is available for 3D interconnection, the employment of Through Silicon Vias (TSVs) is a feasible approach for the interconnection between stacked layers. However, the drawback for current 3D technologies is that TSVs are usually very expensive in terms of silicon area limiting their usage.

This work presents a 3D mesh NoC architecture called Lasio, exploring architectural impacts of 3D versus 2D NoC topologies on latency, throughput, and buffers occupancy. It also analyzes the influence of buffer depth on communication latency and on application latency. Such evaluations considered varied network parameters, such as traffic patterns, buffer depth, TSVs serialization level, and a range of packet sizes. Besides, during this work, it was implemented a TSV serialization scheme on the Lasio NoC, and it was analyzed the impact of such serialization scheme on area cost, power dissipation, network and application latency, and occupancy on buffers of input ports for a 4x4x4 3D mesh NoCs with different serialization degrees.

Experimental results show that, in average, 3D topologies minimize 30% the application latency and increase 56% the packets throughput, when compared to 2D topologies. In addition, this work highlights that when applying an appropriate buffer depth,

the application latency is reduced up to 3.4 times for 2D topologies and 2.3 times for 3D topologies. Additional results demonstrate that NoCs 3D approach reduce the links occupancy when compared to 2D counterpart, which potentially leads to higher throughput and more dissipation power and latency efficiency. Moreover, results also demonstrate that the proposed serialization scheme allows reducing TSVs usage with low performance cost, displaying the potential benefits of the scheme in 3D NoC-based MPSoCs.

Keywords: MPSoCs, 3D NoC, 3D Integrated Circuit, Through Silicon Via (TSV), TSV serialization.

LIST OF FIGURES

Figure 1.	Microprocessor transistor counts (1971-2011) [EMB13].....	15
Figure 2.	Three-dimensional integrated circuit [FEE07].....	17
Figure 3.	3D Network-in-Memory architecture [LI06].	23
Figure 4.	2D and 3D routers architecture [FEE07].....	24
Figure 5.	Clos NoC - 512 nodes configuration [ZIA11].	26
Figure 6.	TSV serialization scheme [PAS09].....	27
Figure 7.	Injection and ejection stages for $k = 4$ layers [RAM09].....	29
Figure 8.	3D NoC with layers half and quarter connected by TSVs [XU10].	30
Figure 9.	3D Mesh NoC with TSV Squeezing [LIU11].	31
Figure 10.	A generic 3D mesh NoC [GHI12a].....	34
Figure 11.	Lasio router architecture [GHI12a].	35
Figure 12.	Example of Lasio signals between routers 121 and 221 [GHI12b].	36
Figure 13.	Packet structure of Lasio NoC [GHI12b].	37
Figure 14.	Example of two simultaneous connections in the router [GHI12b].	38
Figure 15.	The basic structure of a 3D IC constructed from 2D ICs [PAP11].	44
Figure 16.	TSV serialization scheme.	46
Figure 17.	3D NoC design flow.	49
Figure 18.	Electra 3D NoC generation tool.....	52
Figure 19.	General environment setup.....	53
Figure 20.	Communication latency metrics [MOR10].....	56
Figure 21.	NoC and App latencies comparison between Lasio 3D NoC (4x4x4 mesh) and Hermes 2D NoC (8x8 mesh).....	60
Figure 22.	NoC latency versus App latency for different buffer depth and packet sizes.	61
Figure 23.	Traffic influence on NoC latency and on App latency.....	62
Figure 24.	NoC throughput and App throughput behavior according to five sizes of packets nine depths of buffer.	63
Figure 25.	Traffic influence on NoC and on App throughput.	63
Figure 26.	Top ports buffer occupancy.....	65
Figure 27.	Buffer occupancy of top ports (in percentage) for each router of the three lower layers 0 (a), 1 (b) and 2 (c).	66
Figure 28.	Average NoC latency (first column) and relative gains (second column) for the following variations: (a) packet size (10% of injection rate and 8-flit buffer depth); (b) injection rate (16-flit packet size and 8-flit buffer depth); (c) buffer depth (10% of injection rate and 16-flit packet size).....	67
Figure 29.	Average buffers occupancy (first column) and relative gains (second column) for the following variations: (a) packet size (10% of injection rate and 8-flit buffer depth); (b) injection rate (16-flit packet size and 8-flit buffer depth); (c) buffer depth (10% of injection rate and 16-flit packet size).....	69
Figure 30.	Total application latency (first column) and relative gains (second column) for the following variations: (a) packet size (10% of injection rate and 8-flit buffer depth); (b) injection rate (16-flit packet size and 8-flit buffer depth); (c) buffer depth (10% of injection rate and 16-flit packet size).....	70
Figure 31.	Average network latency, measured in clock cycles (a), and relative gain (b) for varying serialization schemes.	71
Figure 32.	Application total latency, measured in clock cycles (a), and relative gain (b) for varying serialization schemes.	72

Figure 33.	Bottom ports average occupancy(a) and relative gain (b) for varying serialization schemes.....	72
Figure 34.	Top ports average occupancy(a) and relative gain (b) for varying serialization schemes.	73
Figure 35.	<i>File_io.out</i> file extracted data.	87

LIST OF TABLES

Table 1.	Related work summary. Shaded positions indicate no data available in the references consulted.	33
Table 2.	Example of Lasio NoC switching table [GHI12b].	38
Table 3.	3D Layers Interconnections Technologies.	43
Table 4.	High-density TSV projections in 2008 ITRS update [PAP11]	45
Table 5.	Standard cells area and power results for lasio synthesis with four serialization schemes. Static power stands for the measured leakage power of standard cells and dynamic power assumes 50% of switching activity.	74

LIST OF ACRONYMS

B2B	<i>Back-to-Back</i>
BEOL	<i>Back-End-of-Line</i>
CMOS	<i>Complementary Metal-Oxide-Semiconductor</i>
D2D	<i>Die-to-Die</i>
D2W	<i>Die-to-Wafer</i>
dTDMA	<i>dynamic Time-Division Multiple Access</i>
F2B	<i>Face-to-Back</i>
F2F	<i>Face-to-Face</i>
FEOL	<i>Front-End-of-Line</i>
FIFO	<i>First In, First Out</i>
Flit	<i>Flow Control Unit</i>
GALS	<i>Globally Asynchronous Locally Synchronous</i>
IC	<i>Integrated Circuit</i>
ITRS	<i>International Technology Roadmap for Semiconductors</i>
LM	<i>Layer-Multiplexed</i>
MPSoC	<i>Multiprocessor System-on-Chip</i>
NI	<i>Network Interface</i>
NoC	<i>Network-on-Chip</i>
OCP	<i>Open Core Protocol</i>
PE	<i>Processing Element</i>
PoP	<i>Package on package</i>
RPM	<i>Randomized Partially Minimal</i>
RTL	<i>Register Transfer Level</i>
SoC	<i>System on Chip</i>
TSV	<i>Through Silicon Via</i>
VHDL	<i>VHSIC Hardware Description Language</i>
W2W	<i>Wafer-to-Wafer</i>

3D NETWORK-ON-CHIP ARCHITECTURAL EXPLORATION

CONTENTS

1	INTRODUCTION	15
1.1	OBJECTIVES	19
1.2	DOCUMENT OUTLINE	20
2	THEORETICAL BACKGROUND	22
2.1	3D AND 2D NETWORKS ON CHIPS TOPOLOGIES COMPARISON	22
2.2	TSV SERIALIZATION ON 3D NETWORKS ON CHIPS	26
3	LASIO 3D NETWORK ON CHIP ARCHITECTURE	34
3.1	LASIO TOPOLOGY	34
3.2	ROUTER ARCHITECTURE AND INTERFACE	35
3.3	PACKET STRUCTURE	36
3.4	PACKET ROUTING	37
3.5	ARBITRATION	38
3.6	SWITCHING	39
3.6.1	<i>Packet Switching</i>	39
3.7	FLOW CONTROL	40
3.7.1	<i>Credit-Based</i>	40
4	THROUGH SILICON VIA (TSV) INTERCONNECTION	42
4.1	TSV-BASED 3D INTEGRATION TECHNOLOGIES	43
4.2	TSV SERIALIZATION SCHEME	45
5	ELECTRA - 3D NOC GENERATION TOOL	48
5.1	ELECTRA FEATURES	49
6	ENVIRONMENT SETUP	53
6.1	TRAFFIC SCENARIOS	54
6.1.1	<i>All-to-All</i>	54
6.1.2	<i>Complement</i>	54
6.1.3	<i>Traffic Scenarios Variations</i>	55
6.2	METRICS EVALUATED	56
6.2.1	<i>Packet Latency</i>	56
6.2.2	<i>Packet Throughput</i>	57
6.2.3	<i>Buffer Occupancy</i>	57
6.2.4	<i>Area Consumption and Power Dissipation</i>	58
7	LASIO PERFORMANCE EVALUATION	59
7.1	2D AND 3D NoC TOPOLOGIES COMPARISON	59
7.1.1	<i>Network and Application Latencies</i>	59

7.1.2	<i>Network and Application Throughputs</i>	62
7.2	BUFFER OCCUPANCY ANALYSIS AND THE SERIALIZATION TECHNIQUE ON THE 3D NoC LASIO	64
7.2.1	<i>Buffer Occupancy</i>	64
7.3	SERIALIZATION EFFECT ON THE LASIO 3D NoC	66
7.3.1	<i>Latencies and Buffer Occupancy</i>	66
7.3.2	<i>Area Consumption and Power Dissipation</i>	73
8	CONCLUSIONS AND FUTURE WORKS	75
8.1	FUTURE WORKS	77
	REFERENCES	79
	APPENDIX A: LASIO 3D NOC REGISTER TRANSFER LEVEL (RTL)	85
	APPENDIX B: REPORT OUTPUT FILES	87

of fact, current applications require some features that when fully implemented on-chip imply much more complex SoCs with tight performance requirements for data communication and computation. Nonetheless, integrating all of these computational resources requires efficient communication resources as well [JAN03]. Additionally, one of the major problems associated with SoC designs arises from non-scalable global wire delays. In spite of global wires carry signals across a chip, such wires usually do not scale in length with technology scaling [HOR01]. In this sense, the rise in scale and the complexity of SoC design is hampered by the communication challenges among the internal circuits. Such challenges might be observed in several design requirements, such as modularity, flexibility, wire efficiency, scalability, performance, efficient distribution of clock signals, and rational balancing between gate delays and wire delays [BEN02].

In this context, Network on Chip (NoC) has emerged as a SoC paradigm for multicore SoC design facilitating modularity by defining a standard interface, increasing reliable operation of interacting SoCs components, improving flexibility, and reaching higher bandwidth [BEN02][DAL01]. NoCs have been proposed as promising solution to deal with the global communication problem of complex SoCs, which faces huge design challenges and as up-and-coming packet-based communication architecture for Multiprocessor System on Chip (MPSoC) design, especially due to efficient energy consumption, increased scalability and throughput [JAN03]. Nevertheless, increasing the number of cores over a 2D (Two-Dimensional) plane might present drawbacks, such as, NoC efficiency reduced due to long NoC diameter, overall communication distance and wire delay, latency and power dissipation growth. According to the International Technology Roadmap for Semiconductors (ITRS), new interconnection paradigms are essential for the next few years [ITR12] and one direction is to extend existing 2D tile-based MPSoC architectures [RAM09] into three dimensions.

3D (Three-Dimensional) ICs can be dated back to as early as 1980s [KAW83] [NAK84], although, only in recent years, 3D IC has become a hot topic in both academia [DON09] and industry [TOP06], which attests the ITRS statement. 3D ICs are currently seen as one of the most interesting technologies for scalability, power and performance demands of next generation of SoCs. In 3D integration technology, where multiple layers of active devices are stacked above each other and vertically interconnected, as shown in Figure 2 [FEE07], the interconnection power is reduced by eliminating long global wires

and reducing off-chip I/O transactions, while providing low-latency interconnections between stacked heterogeneous circuits [GRA11].

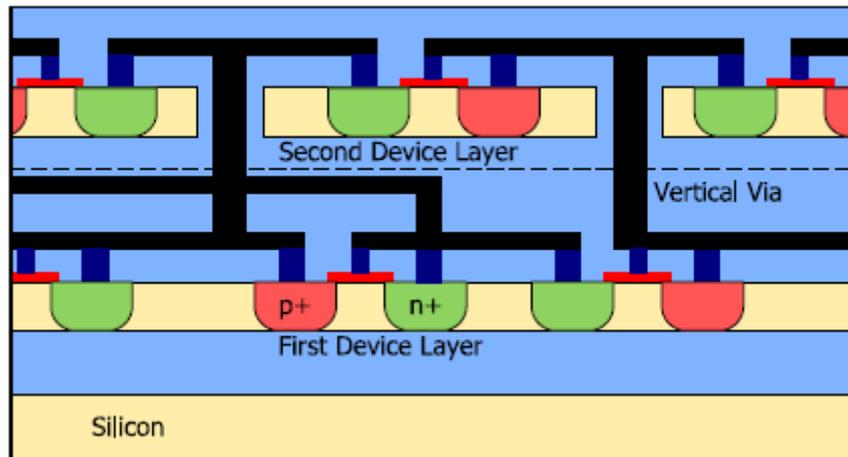


Figure 2. Three-dimensional integrated circuit [FEE07].

Additionally, 3D enables to implement multiple smaller dies generally showing higher yields than one large die of the equivalent 2D implementation with nearly the same total area, and allowing a reduction of design effort and cost [MIL13]. Indeed, 3D integration has attracted significant attention in recent years providing opportunities for chip architecture innovations, improving communication bandwidth and energy efficiency in multicore architectures (IBM and Tezzaron have presented promising results and test chips with 3D IC technology [PAT06]). Furthermore, 3D stacking allows heterogeneous systems/subsystems integration, which may require processing steps in current single-die technologies [SUN05] [WOL08], and it provides new applications range of complex system architectures [LOI11]. 3D ICs allow more performance enhancements with less scaling concerns [RAH10], providing less noise sensible systems [PAT06], and enable silicon reuse. The expectation is to solve major issues such as external memory pressure and latency whilst maintaining reasonable power dissipation.

As architectures based on NoCs are getting increasingly technical consensus, 3D NoCs, which combine the benefits of short vertical interconnections of 3D ICs and the scalability of NoCs, have become an innovative technology by integrating NoC systems in a 3D manner. Moreover, 3D NoCs have potential benefits, including smaller chip paths, higher transistor density, shorter wiring delays and the considerable reduction in the length and number of global interconnections [PAV06]. These benefits enable 3D NoC structures to offer higher performance, improved packaging density, and lower interconnection power dissipation to SoCs compared to their 2D counterparts. Another interesting characteristic is

that 3D NoC approach offers a matchless platform to implement Globally Asynchronous Locally Synchronous (GALS) design paradigm [MUT00]; this makes the clock distribution and timing closure problems more manageable and enables 3D technology to be suitable for heterogeneous integration.

Nevertheless, those benefits previously mentioned can only be achieved with some additional costs. It is recognized that vertical links are normally considered a bottleneck in 3D NoC design, mainly for implying links with more area and sometimes mechanisms to serialize and deserialize the communication. Many bonding technologies like Die-to-Die (D2D), Die-to-Wafer (D2W), and Wafer-to-Wafer (W2W) have been investigated and analyzed recently. Out of these, W2W is an interesting and inexpensive implementation technology for 3D ICs [PAV08]. It relies on Through Silicon Via (TSV) use [SPI04] for vertical connectivity. TSV has become an interesting and viable solution to stack several 2D IC layers together [PAV08] providing vertical connections between different stacked dies and guaranteeing short wire length with low capacitive load and, hence, providing fast connections between two or more chip layers. Moreover, comparing vertical interconnections that employ TSVs in 3D NoCs to other technologies, such as wire bonding and metal bumps, TSVs are able to provide higher bandwidth. Hence, TSVs have become a major choice to replace them [BAN01].

Regarding TSV technology be relatively immature, 3D ICs suffer from higher failure rates comparing to traditional 2D ICs, especially when considering the huge number of TSVs required [YIN11]. Moreover, some aspects as data traffic through TSVs might become a bottleneck in 3D NoCs, since TSVs have much bigger pathways than those on-chip, and very different area and impedance [LIU11]. Another important aspect is the area footprint of TSVs in each layer, which is no longer negligible, since each TSV requires a pad for bonding to a wafer layer (details in Table 4).

The TSV serialization is a technique that tends to reduce area, power dissipation and the number of vertical connections in the 3D technology. In this sense, a TSV serialization scheme is a promising approach, enabling higher design space exploration to reduce the area cost of 3D NoCs. Hereupon, the technique can help coping with technological issues, especially the limited number of TSVs in a 3D integration system.

1.1 Objectives

The present work aims to implement, functionally validate and explore a 3D NoC architecture called Lasio, which is an extension of Hermes 2D NoC [MOR04], exploring aspects regarding a 3D NoC architecture and evaluating its overall performance. The purposes of this work might be analyzed from two angles: objectives and contributions.

Strategic objectives:

- To understand and analyze 3D ICs as a new trend of system integration;
- To explore 3D NoC technology and architecture;
- To implement a 3D NoCs generation tool;
- To evaluate the benefits and drawbacks of 3D NoCs regarding architectural aspects;
- To analyze the TSV impact as a new technology implemented for vertical communication channel between layers.

Specific Contributions:

- The first contribution is the Electra implementation, which is a generation tool for 3D NoCs that was indispensable and essential for this work development. Electra allows generating different 3D NoCs varying their structures and architectural configurations, which guarantee a wide network exploration and evaluation.
- The second contribution is to analyze the architectural impact of 2D and 3D NoC topologies on network and application latencies and data throughput comparing Lasio 3D NoC with Hermes 2D NoC. The performance evaluation shows the significant impact of buffer depth on application latency for 2D and 3D topologies, which cannot be neglected in NoCs design exploration;
- The third contribution focus on assessing the significant impact of buffer depth and packets size on the Lasio 3D NoC regarding application and network latencies and throughput;
- The fourth contribution is to analyze buffers occupancy on the Lasio. Simulation results demonstrate that NoCs 3D approach reduce the links occupancy when compared to 2D counterpart, which potentially leads to higher throughput in the NoC and more power and latency efficient systems.

In addition, results suggest the possibility of TSV serialization without performance degradation at system level;

- The fifth contribution is to propose and analyze a TSV serialization mechanism on the Lasio 3D NoC evaluating network and application latency, as well as buffers occupancy of input ports under controlled execution of test sets. The obtained results suggest that the serialization scheme reduces the TSV usage without significant degradation in the NoC overall performance and can be scalable for different application sizes;
- The final contribution still focuses on TSV serialization scheme in order to enhance overall 3D NoC performance through a reduction on buffers occupancy. Results point to a tradeoff between serializing TSVs and dynamic power. Besides, the proposed scheme presents good scalability for different serialization levels, especially for the TSVs serialization degrees evaluated in terms of area and static power;

Such evaluations allow NoC performance assessment under controlled executions of test sets. Those tests are composed of a couple of synthetic scenarios, varying aspects such as application size, injection rate¹, buffer depth, packet size, flit² size and TSV serialization degree.

1.2 Document Outline

The remainder of this work is structured as follows. Chapter 2 discusses related works on 2D versus 3D NoC design concerning application requirements, input traffic, set of experiments, and on 3D IC technology and the TSV impact on 3D NoC design, pointing some TSV serialization structures. Chapter 3 describes the architecture of Lasio, which is the 3D NoC mesh implemented and used for the experimental purposes. Chapter 4 characterizes the TSV technology and the possibility of their use in 3D systems integration

¹ Clock cycles percentage rate in which the packets are being injected into the network (e.g., if the injection rate is 25%, it means that in 75% of the total clock cycles the network is not receiving any packet).

² Flit is the minimal logic unit used for flow control.

through a proposed serialization scheme. Chapter 5 presents a 3D NoC generation tool named Electra that was developed in the context of this work with the intention of design 3D NoCs with different and parameterizable configurations. Chapter 6 presents the experimental setup. Chapter 7 discusses Lasio 3D NoC performance evaluations. Finally, Chapter 8 concludes the work contribution and highlights some possible future works.

2 THEORETICAL BACKGROUND

This chapter surveys the state-of-the-art regarding two main topics – *3D and 2D NoC Topologies Comparison* and *TSV Serialization on 3D NoC*. As mentioned before, 3D NoCs have become an innovative technology enabling to offer higher performance and lower interconnect power dissipation applied especially in the integration of a SoC. The following sections discuss 3D NoC architectures design and their main characteristics, positioning the present work in such perspective.

2.1 3D and 2D Networks on Chips Topologies Comparison

Current MPSoCs are normally implemented targeting 2D communication architectures. However, in the last few years, there has been a growing interest in 3D ICs from academia and industry to alleviate the interconnect bottleneck problem faced by 2D ICs. In this way, IBM [BER07] has stated: “Improvements in on-chip wire delay and in the maximum number of I/O per chip have not been able to keep up with transistor performance growth” when computer chips remain essentially 2D. Both IBM [BER07] and Tezzaron [PAT06] presented promising preliminary results and test chips on 3D IC technology applied to SoC devices. However, 3D technology may also bring new design challenges and some drawbacks. In this sense, researches comparing 3D NoC topologies with their 2D counterpart have been proposed. In addition, studies that analyze and evaluate 3D integration are attracting researches in order to point and clarify benefits and challenges. The following related works describe some of these efforts.

Li et al. [LI06] focus their work on 3D chip multiprocessor design and memory networking issues, especially in the context of data management in large L2 caches, analyzing the challenges for L2 design and management. As first contributions, the authors propose a router architecture and a topology design that makes use of a network architecture embedded into the L2 cache memory. They show that a 3D architecture with no dynamic data migration generates better performance than a 2D architecture that employs data migration. The proposed authors’ architecture for improving the performance of multiprocessors chips with large shared L2 caches involves placement of CPUs on several layers of a 3D chip with the remaining space filled with L2 cache banks. Besides, they suggest the use of dynamic Time-Division Multiple Access (dTDMA) vertical buses as “communication pillars” between the layers (Figure 3). These buses are capable to

produce one single hop in the communication among the layers because of the short distance between them and they can be interfaced to a traditional NoC router for intra-layer traversal.

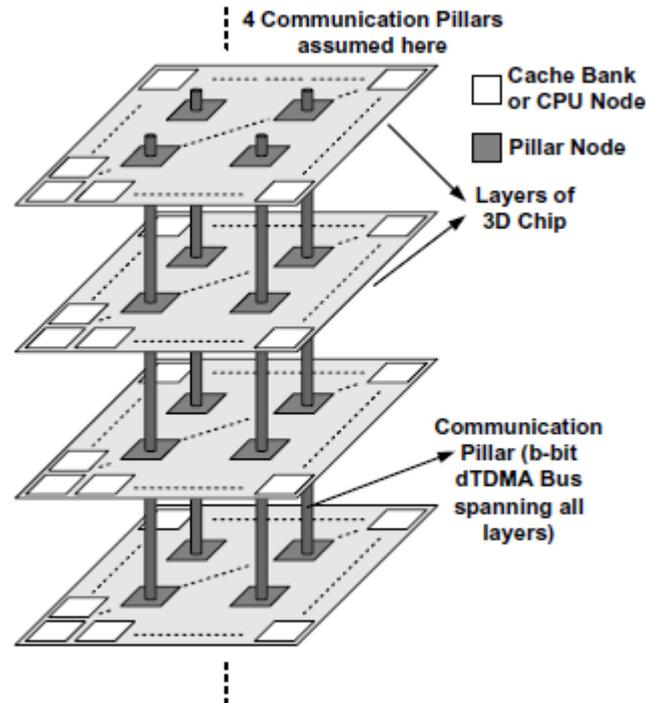


Figure 3. 3D Network-in-Memory architecture [LI06].

Such hybridization of buses and networks to provide the interconnect fabric between CPUs and L2 caches, with a common NoC router and a bus link in vertical dimension, utilizes a valuable attribute of 3D chips - the very small distance between the layers. The hybrid system allows single-hop communication between nodes connected by the vertical bus, and provides both performance and area benefits. Furthermore, hybridization between the NoC router and the bus requires only one additional link (instead of two) on the NoC router. Li et al. have shown that the proposed 3D architecture reduces average L2 access latency significantly over 2D topologies. Additionally, the authors have demonstrated that the bandwidth of the vertical interconnections has a significant impact on the L2 cache latencies. However, this analysis pertains only to multiprocessors chips and does not consider the use of 3D network structures for application-specific SoCs. Moreover, under adverse traffic conditions, e.g., when source and destinations are on different layers, the shared bus might present itself a throughput bottleneck, which makes this architecture feasible although it is not a consensus. In addition, not all NoC routers can include a vertical bus due to technological limitations and especially due to router complexity issues.

Feero et al. [FEE07] evaluate the performance of a variety of 3D NoC architectures compared to existing 2D counterparts. They claim that 3D NoCs are capable of achieving higher throughput, lower latency, and lower energy consumption at the cost of small silicon area overhead. The authors compared the following topologies: (i) 2D Mesh versus 3D Mesh; (ii) 2D Torus versus 3D Torus; and (iii) Stacked Mesh versus Stacked Torus. Such evaluations were performed using realistic traffic patterns in terms of standard metrics. In order to evaluate such topologies, the authors implemented a typical router for stacked architectures, which employs 7 ports for communication and contains one link to the IP block, 2 links in the x-dimension, 2 links in the y-dimension, and one link to the bus, as shown in Figure 4. This eliminates one of the ports for the z-direction compared to a standard 3D mesh or torus. Such port minimization will cease to see speed benefits due to contention issues over the bus at a certain size in the vertical dimension.

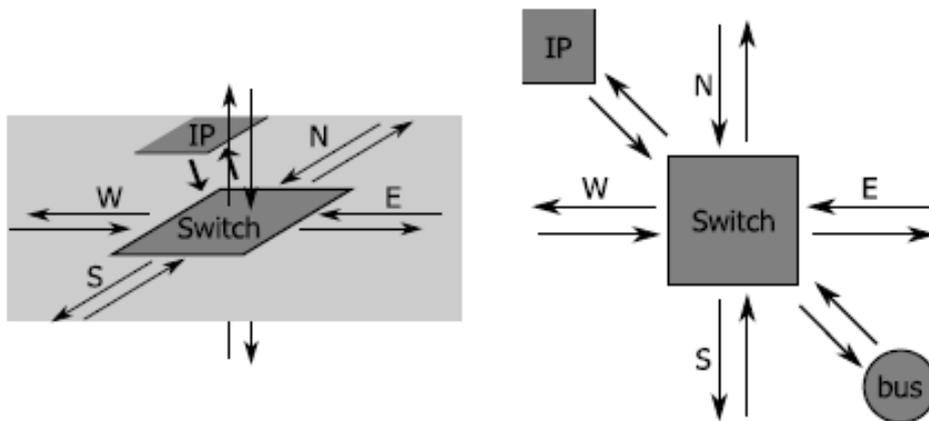


Figure 4. 2D and 3D routers architecture [FEE07].

Feero et al. have also shown that 3D NoCs have more switches that are complex but offer better performance and lower energy consumption for communication. They demonstrated that besides reducing the footprint in a fabricated design, 3D network structures provide better performance compared to traditional 2D NoC architectures. Feero and Pande [FEE09] conducted similar analyses. The authors have used synthetic localized and uniform traffic to evaluate the performance of 3D NoC architectures extending previous work to comprise 3D tree-based NoCs topology. Their study demonstrates the 3D NoC superior functionality in terms of throughput, latency, energy consumption, and wiring area overhead when compared to 2D counterpart implementation.

Pavlidis and Friedman [PAV07] compared 2D mesh structures with their 3D counterparts by analyzing analytic models for the zero-load latency³ and power dissipation with delay constraints of these networks that capture the effects on the 3D NoC performance caused by different topologies. This is an evaluation that shows some of the advantages of 3D Networks on Chips - a performance improvement of 40% and 36% and a decrease of 62% and 58% in power dissipation is demonstrated for 3D NoC as compared to a traditional 2D NoC topology for a network size of $N = 128$ and $N = 256$ nodes, respectively. The present work has some differences from the analysis in [PAV07]. The latter based its experiments using analytic traffic models varying the number of nodes in the architecture while the former presented several different synthetic scenarios and parameters such as injection rate, flit size and buffer depth being changed and evaluated. Moreover, [PAV07] neither applies any real traffic pattern, nor measures other relevant performance metrics.

Other works focus on 3D NoCs implementation and evaluation. Park et al. [PAR08] have developed MIRA, which is a 3D mesh NoC topology. The authors propose an on chip network based on a 3D multi-layered router (3DM) that is designed to be stacked across the multiple layers of a 3D chip. Such router contains additional channels used to improve the communication. Besides, the 3D router design does not need additional functionality as compared to a 2D router and only requires distribution of the functionality across multiple layers. Experiments with random uniform traffic and seven real workloads proved the efficiency of MIRA to minimize the NoC temperature, and to reduce the energy consumption and latency when compared to traditional 2D and 3D mesh NoCs, achieving up to 42% reduction in power dissipation and up to 51% improvement in average latency with synthetic workloads. Zia et al. [ZIA11] evaluated the energy consumption on large MPSoCs for several NoC topologies, concerning 2D NoCs and their 3D counterparts. They also implemented a 3D Clos NoC (CNoC) with 3D integration as viable network topology for many cores CMPs. To evaluate CNoC and others topologies (fat tree, flattened butterfly, and mesh) in terms of network scalability and energy efficiency, they used

³ The zero-load latency of a network is the latency where only one packet traverses the network. Although such a model does not consider contention among packets, the zero-load latency may be used to describe the effect of a topology on the network performance.

experiments that take into account the influence of the number of nodes and some router types on the minimization of energy consumption and latency. In addition, the NoC energy consumption was evaluated according to flit size and injection rate. They conclude for all experiments that CNoC is more energy and latency efficient than the 2D counterpart when the network size is scaled to 512 nodes. Such scaled network is illustrated on Figure 5 [ZIA11] where there are eight blocks interconnected containing 64 nodes each one. The routers forming the middle three stages are denoted as CM1, CM2 and CM3, respectively.

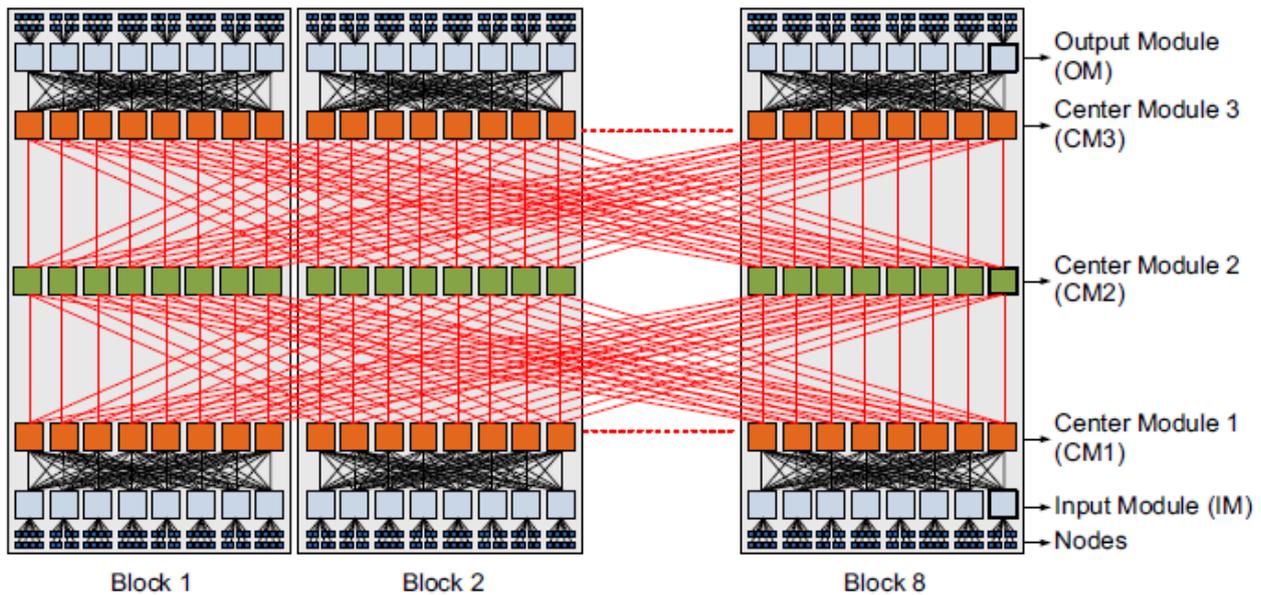


Figure 5. Clos NoC - 512 nodes configuration [ZIA11].

Nevertheless, [PAR08] assumes the processor cores are designed in 3D, which makes it difficult to reuse existing highly optimized 2D processor core designs. In addition, both [PAR08] and [ZIA11] do not consider possible TSV communication bottlenecks, which might affect latency, throughput and some others parameters and their architectures, Mira and CNoC, do not contemplate TSV serialization in any level.

2.2 TSV Serialization on 3D Networks on Chips

As mentioned before, long interconnections utilized on traditional 2D ICs can be replaced by much shorter vertical TSV interconnections in 3D ICs. However, TSVs pads distributed on each layer might represent some challenges on power density and routing congestion. In this sense, serialization of vertical TSV interconnections in 3D ICs is proposed as one way to address these challenges. Therefore, efforts to investigate design techniques and methodologies to exploit the benefits of 3D technologies are recurrent.

Several publications discuss the idea of using TSV as vertical communication links between layers, and propose TSV serialization schemes, in order to reduce the interlayer links, alleviating the bottlenecks related to interconnect problems. Below follows a revision of some relevant related works regarding such subject.

Pasricha [PAS09] has proposed a serialization scheme of serial vertical TSV interconnections in 3D ICs to address challenges regarding TSV area footprint and power dissipation. The author sustains that such serialization of TSV interconnections have the benefit of reducing the number of TSV interconnections and line drivers, which in turn reduce the TSV interconnect area footprint in each layer. Additionally, since TSV density is limited by fabrication cost factors, fewer interconnect TSVs can make way for more thermal TSVs, which will lead to more thermal-efficient IC designs.

The TSV serialization scheme proposed is centered on a shift-register based serialization scheme. Figure 6 illustrates this scheme, where Figure 6(a) shows the block diagram of the transmitter (or serializer) at the source. When a word becomes available for transfer in the transmission buffer, the RS flip-flop is enabled, thereby enabling the ring oscillator, which generates a local clock signal.

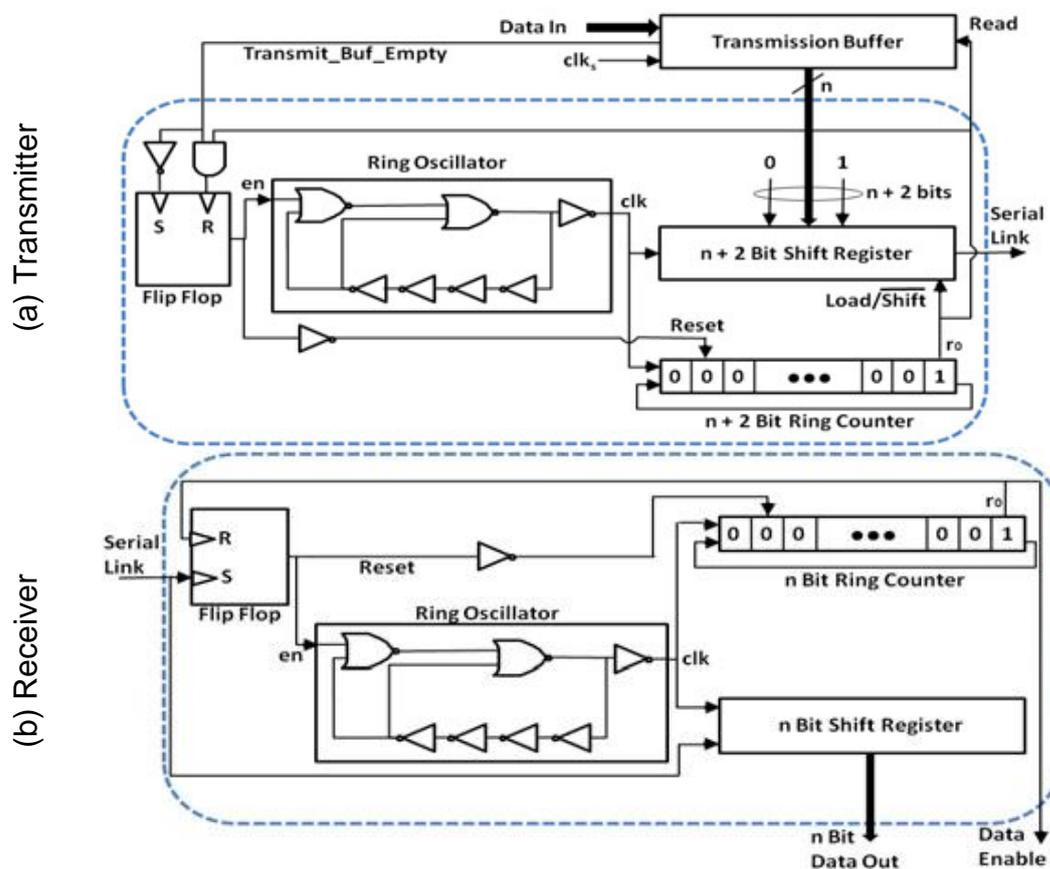


Figure 6. TSV serialization scheme [PAS09].

At the first positive edge of this clock, an $n+2$ bit data frame is loaded in the shift register. In the next $n+1$ cycles, the shift register shifts out the data frame bit by bit. Then the next data word is loaded into the shift register from the transmission buffer on the next positive clock edge. Figure 6(b) presents the block diagram of the receiver (or deserializer) at the destination. The RS flip-flop in the receiver is activated when a low-to-high transition is detected on the input serial line. After being activated, the flip-flop enables the receiver ring oscillator and the ring counter. The n -bit data word is read bit by bit from the serial line into a shift register, in the next n clock cycles. Thus, after n clock cycles, the n -bit data will be available on the parallel output lines. Then the receiver is ready to start receiving the next data frame.

In experimental results of [PAS09], Pasricha shows that a 4:1 serialization of TSV can save up to 70% of TSV's area. However, significant performance overhead occurs in this approach caused by serial-parallel conversion, such that power dissipation increases and performance reduces even more as the degree of serialization becomes higher.

Ramanujam and Lin [RAM09] propose a Layer-Multiplexed (LM) architecture of 3D NoCs, which takes advantage of the short interlayer wiring delays enabled by 3D technology. The LM architecture proposes to replace the one-layer-per-hop routing in a conventional 3D mesh with vertical demultiplexing and multiplexing structures (Figure 7) [RAM09]. At packet injection, flits are *demultiplexed* uniformly to the k layers using the *packet injection stage*. Once *demultiplexed* to a horizontal layer, flits are routed on this plane using either minimal XY or YX routing with equal probability. At the destination (X, Y) coordinates, packets from all layers are *multiplexed* at the destination processor in the *packet ejection stage*. Moreover, LM exploits a routing algorithm called Randomized Partially Minimal (RPM)⁴.

Both architectures, mesh and LM were evaluated on two different sizes - $8 \times 8 \times 4$ and $4 \times 4 \times 4$. Comparing the LM with a conventional 3D mesh, the former dissipates 27% less power, attains 14.5% higher average throughput and achieves 33% lower worst-case hop count on a $4 \times 4 \times 4$ topology. However, the benefits regarding throughput and hop count

⁴ RPM first routes in the Z dimension to a randomly chosen intermediate XY plane. It then routes flits on each XY plane using either minimal XY or YX routing with equal probability. Finally, it routes flits to their final destinations along the Z dimension.

(which influences latency) that authors claim are only achieved when 4x4x4 topology is used and for one specific traffic pattern. In fact, when evaluating larger topologies (e.g., 8x8x4) and different traffic scenarios, those metrics have no improvements whatsoever or have worse results.

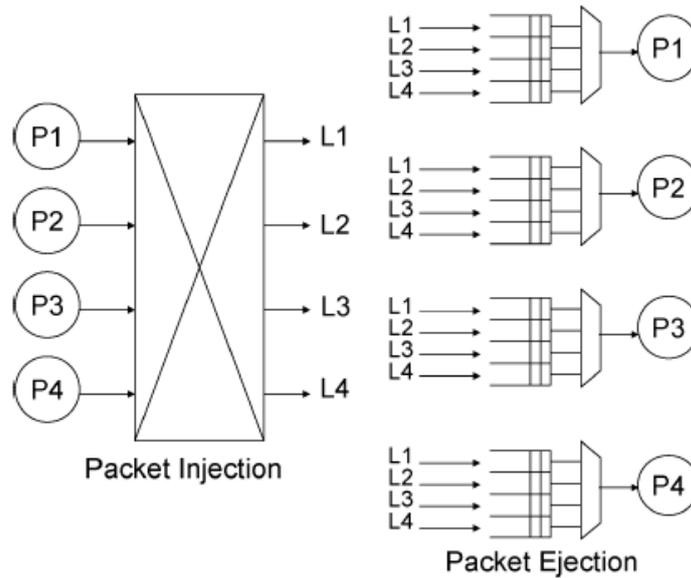


Figure 7. Injection and ejection stages for $k = 4$ layers [RAM09].

Sun et al. [SUN10] propose a technique on the vertical link design - referred as “3D quasi-serial physical link” - to exploit the high bandwidth offered by TSVs replacing 3D interconnections by a serialization scheme using a synchronous TSV. Such link architecture is designed as four serial links placed in parallel and acting as one quasi-serial link (with the same capacity of the four serial links) for NoC. Besides, clock TSVs are shared among the four serial links. Therefore, the authors increase the redundancy of clock TSV to four.

The proposed scheme by [SUN10] achieves five times less area than the traditional synchronous parallel link (transmits data-words of fixed size along with the clock required for synchronization). However, only the serialization of one link is covered. There is no serialization of multiple links, which may jeopardize the results and consequently the proposed scheme.

Xu et al. [XU10] presented a study about the impact of TSVs on 3D NoC design with five layers and analyzed both performance and manufacturing cost for different TSV quantities and placements. Their analysis explored three designs with full, half, and quarter layer-layer connections between layers. Figure 8 [XU10] illustrates half and quarter

placement strategies. The TSV placement choices were based on the lowest average hop count possible for half and quarter pillars configuration. The authors showed that full TSV connections outperformed others in terms of network latency and network power dissipation, mainly due to the traffic contention between layers are alleviated when more TSVs are used. However, manufacturing cost becomes a tradeoff for this power-performance improvement.

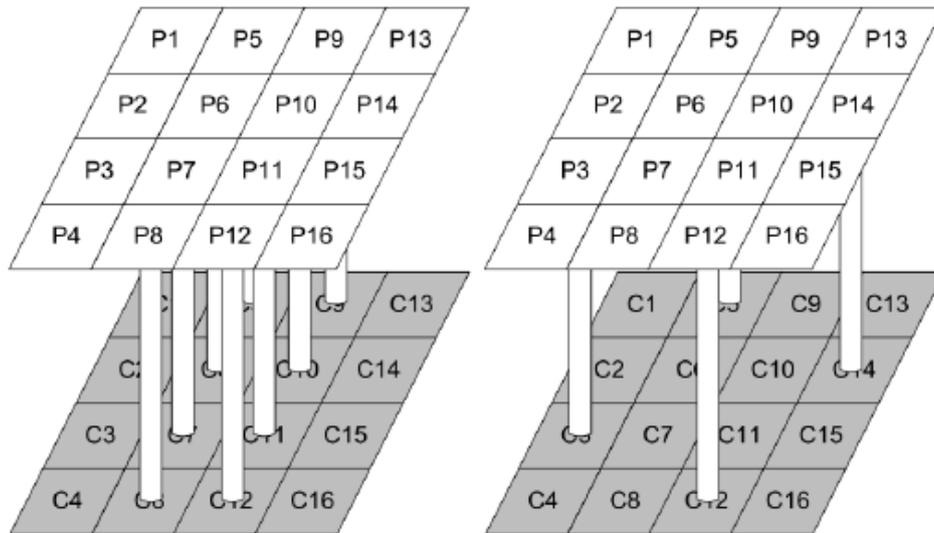


Figure 8. 3D NoC with layers half and quarter connected by TSVs [XU10].

Buttrick et al. [BUT11] show a method to increase the effective number of inter-die connections in 3D ICs to mitigate limitations such as the number of TSVs on a silicon area. In the proposed solution two signals are multiplexed and sent over a single TSV in one clock cycle with very low overhead. The solution relies on both positive and negative edge-triggered flip-flops to capture signals that had been serialized on the TSVs by the system clock. They realize that because of the low overhead incurred by this design, the serialization technique may be used freely on every TSV without concern for an increase in the design area or power consumed by TSVs.

Liu et al. [LIU11] propose a TSV serialization scheme (squeeze adjacent TSVs together) among neighboring NoC routers in a 4x4x2 3D symmetric mesh structure in order to share the same vertical channel. The general idea is that adjacent NoC routers share vertical links based on the observation that TSVs utilization in a 3D symmetric mesh NoC is quite low, and adjacent routers rarely require data transmission in their vertical channels at the same time. Figure 9 shows the proposed squeezing scheme in which four nodes share a single TSV. Whenever there is data transmitting from one layer to the other, it will have to

acquire grant of TSV sharing logic at first. Then data with grant will go straight to TSV sharing logic in current layer and flow through TSV later. Finally, the data will be forwarded to corresponding router from TSV sharing logic in target layer.

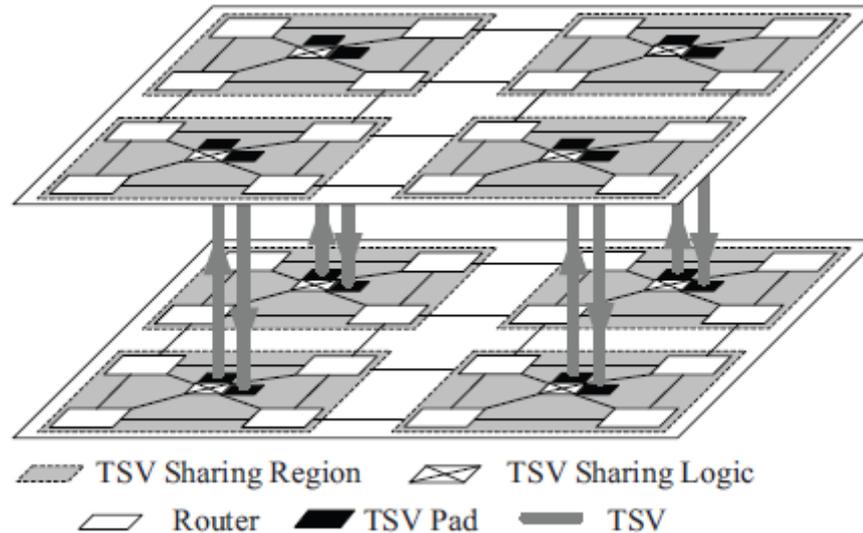


Figure 9. 3D Mesh NoC with TSV Squeezing [LIU11].

The proposed solution can achieve an interesting save on TSV area; however, it has an overhead on network performance. Comparing [LIU11] and the present work, the former proposes a serialization scheme that explores TSV serialization of four routers into a single one. This approach implies that the communication need for each router affects the communication rate of other routers in the same group. In our approach, serialization is performed for vertical links at each router. Thus, it does not directly affect other routers' communications.

Yong et al. [YON11] proposed a 3D NoC design technique that decreases the number of TSVs by grouping communication packets to prevent critical traffic and to reduce the overhead drawbacks of TSVs such as area and cost. Besides, for inter-layer communication a broadcasting-type vertical communication algorithm (the layer which possesses a token is the only one able to broadcast the packets to any other layers) is suggested. The authors reached 39.5% on chip area saving while the performance decreased 8.1% in a 4x4x3 3D mesh NoC with 8 TSVs (four cores grouped into a single virtual group and one TSV is allocated for each group). Nevertheless, there is no control regarding which core has priority on TSVs use, by what means that bottleneck in vertical transmission may occur to affect adversely the overall performance when the network resources increase.

Rahmani et al. [RAH13] proposed a power and area efficient 3D NoC architecture based on power-aware Bidirectional Bisynchronous Vertical Channels (BBVC) to exploit bidirectional bisynchronous vertical channels as a solution to mitigate challenges in 3D NoC design, such as high peak temperatures, power densities and area footprints of vertical interconnections in each layer. The authors claim that utilizing a dynamically self-configurable BBVC, it enables a system to benefit from low-latency nature of the vertical interconnections. Besides, the main idea of the proposed 3D NoC system is to exploit a bidirectional channel for interlayer communication operating at a higher frequency compared to intra-layer communication (the authors have assumed to be two times faster) and being capable of dynamically changing the channel direction between routers in neighboring layers based on the real time bandwidth requirement. It is noteworthy that for the desired system, a bisynchronous FIFO for Up and Down ports of each 3D router is needed to enable vertical channels to operate at different speeds.

This scheme can reduce the number of TSVs and mitigate high power densities and peak temperatures. Simulation results show that the proposed architecture can reduce up to 47% of TSV area footprint and up to 18% NoC power dissipation with a performance degradation. However, replacing a pair of unidirectional vertical channels by a bidirectional one might increase input buffers congestion because of high competitions, which leads to drops in link utilization for some directions.

This master's study extends all previous works (summarized on Table 1) in order to analyze some specific points in 3D NoC architecture:

- Investigate architectural impacts of 2D and 3D NoC topologies and compare both NoCs overall performance;
- Evaluate buffers exploration according to traffic characteristics. It is shown that 3D topologies are able to achieve better performance than their 2D counterpart, but most importantly it is demonstrated how a 3D architecture could be improved according to packets size and traffic behavior;
- Analyze Lasio 3D NoC under different configurations focusing on a serialization scheme for vertical interconnections and its impact on network and application latency, on the occupancy of input buffers, on area consumption as well as on overall power dissipation.

Table 1. Related work summary. Shaded positions indicate no data available in the references consulted.

3D and 2D NoCs Topologies Comparison

Work	NoC	Requirements	Traffic	Experiments
[LI06]	3D hybrid NoC-bus	reduction of latency and energy consumption	cache migration algorithm	number of TSVs and layers
[FEE07]	3D and 2D Mesh; 3D and 2D Torus; Stacked Mesh and Torus	reduction of latency, and energy consumption and throughput increase	self-similar distribution	injection rate
[PAV07]	3D and 2D mesh	reduction of energy consumption and latency	analytic models	number of nodes
[PAR08]	3D and 2D mesh	reduction of temperature, energy consumption and latency	random and seven real applications	injection rate
[FEE09]	3D and 2D mesh	reduction of latency, wiring area and energy consumption, and throughput increase	several uniform and localized traffic	injection rate, number of 3D layers
[ZIA11]	3D and 2D clos, mesh, ftree, bfly	reduction of latency and energy consumption	uniform using Bernoulli process	number of nodes, injection rate, flit size
[This]	3D and 2D mesh	reduction of NoC and application latency, and NoC and application throughput increase	all-to-all and complement	packet size and buffer depth

TSV Serialization Assessment on 3D NoCs

Work	NoC	Requirements	Traffic	Experiments
[PAS09]	3D mesh	reduction of TSV area and power dissipation	each CMP app has different traffic	TSV serialization and CMP applications
[RAM09]	3D mesh and LM	reduction of power dissipation and worst-case hop count, and throughput increase	several	TSV serialization and 8x8x4 and 4x4x4 topologies
[SUN10]	3D mesh	reduction of area		TSV serialization and number of TSVs
[XU10]	3D mesh	reduction of network latency and power dissipation		number and placement of TSVs and TSV design connections
[BUT11]		reduction of power dissipation and design area		TSV serialization, number of TSVs
[LIU11]	3D mesh	network latency reduction and throughput increase	uniform, shuffle and hotspot	TSV serialization, arbitration schemes and injection rate
[YON11]	3D mesh	area reduction and throughput increase		number of TSVs
[RAH13]	3D mesh and DFS-BBVC-3D-NoC	reduction of area, power dissipation and latency	uniform, NED and hotspot	number of TSVs, communication frequency levels
[This]	3D mesh	reduction of NoC and application latency, buffer occupancy, area and power dissipation, and throughput increase	all-to-all and complement	TSV serialization, injection rate, packet size, buffer depth, application size and flit size

3 LASIO 3D NETWORK ON CHIP ARCHITECTURE

Lasio is a 3D mesh NoC architecture developed during this work, which was based on Hermes 2D NoC [MOR04]. Lasio has the same mechanisms and resources of Hermes, supporting more ports at each router, TSV interconnections between layers and a 7x7 crossbar to enable 3D communication instead of 5x5 usually implemented in the 2D case. Lasio characteristics are described in the subsequent sections.

3.1 Lasio Topology

NoC topology is defined by the connection structure among routers. A direct topology is the one where each router has a set of bidirectional ports linked to other routers, and one port linked to a local Processing Element (PE). The straightforward extension of the 2D mesh structure is a 3D mesh NoC [FEE09], which adds two additional physical ports to each router, one for top and other one for bottom.

Figure 10 [GHI12a] illustrates a generic 3D Lasio mesh NoC. Each layer can have multiple PEs, such as memories and processors. Interlayer communication channels are composed of TSVs that cut across thinned silicon substrates to build connectivity after die bonding. Lasio NoC design uses direct topology, where each tile contains a single PE connected to the NoC by a router local link. The direct topology facilitates the placement of routers and PEs, as well as the routing channels between routers, simplifying the routing algorithm implemented in the control logic. Besides, each router of Lasio has a unique NoC address expressed in XYZ coordinates, and a different number of ports, depending on its position with regard to the NoC limits.

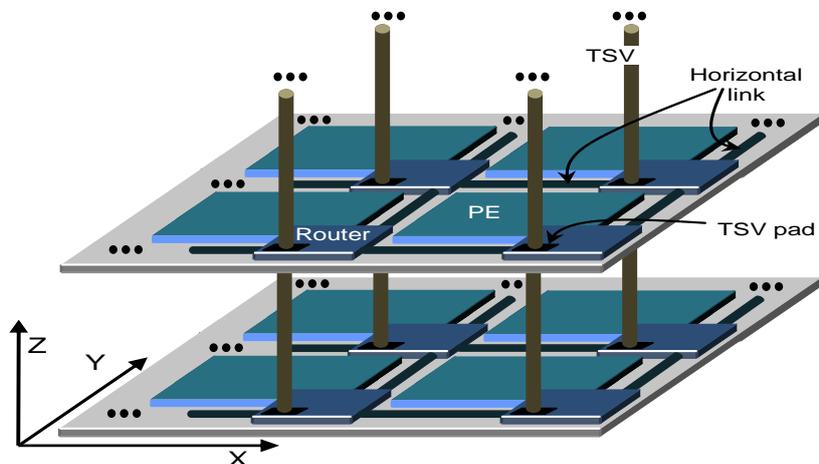


Figure 10. A generic 3D mesh NoC [GHI12a].

3.2 Router Architecture and Interface

Router is the main NoC component responsible for end-to-end communication between PEs. Thus, it must be designed aiming not impact the final SoC area, but also having small energy consumption and low switching time, in order to respect the design requirements. Buffer is one of the main components that influence those two last concerns. Therefore, the data storage strategy is a predominant factor during the NoC design [MAR05].

Figure 11 illustrates the three basic modules that implement the Lasio router: (i) an input buffer for each one of the seven ports working as circular *First In, First Out* (FIFO); (ii) a switch control logic; and (iii) a 7×7 crossbar responsible for ports switching. Five ports are dedicated to intra-layer connections (Local, North, South, East and West). Two other ports (Top and Bottom) ensure the communication between adjacent layers. The Local port establishes a communication between the router and its corresponding PE, while the remaining ports are connected to neighboring routers. Each communication port includes input and output channels, and each input port has a buffer for temporary data storage working as circular FIFO with configurable size and depth. These buffers are used when other packets congest the routing path. In addition, the crossbar module indicates which ports are connected verifying data to be transmitted and the ports availability.

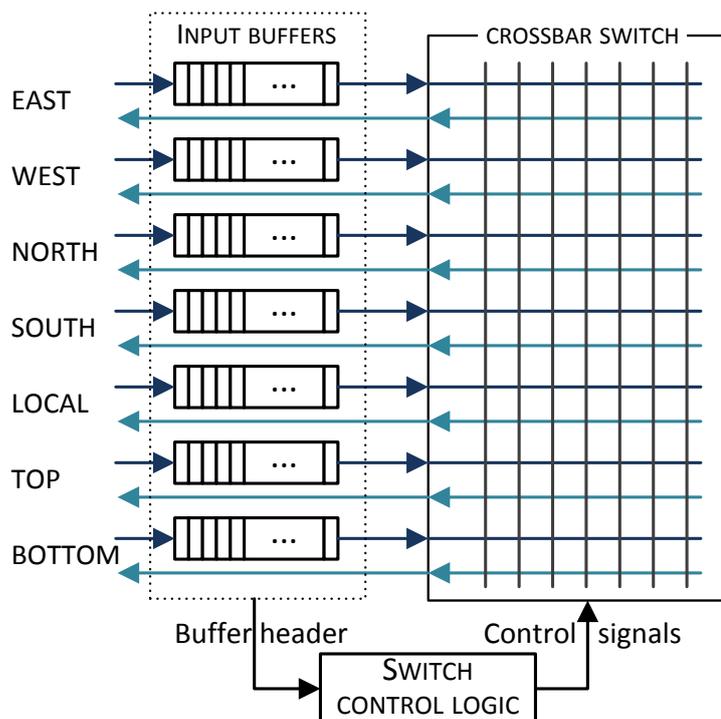


Figure 11. Lasio router architecture [GHI12a].

When an input buffer receives the first flit of a packet, it sends the flit to the switch control logic that executes the arbitration. If the incoming packet request is granted, it performs the routing algorithm, connecting the input port data to the correct output port. If the chosen output port is busy, subsequent flits are stored in the input buffer, and the request remains active until the connection with the output port is established.

Figure 12 shows a bidirectional link between two routers. The output port is composed of the following signals: (i) *clock_tx* that synchronizes data transmission; (ii) *tx* that controls the data availability; (iii) *data_out*, which is a data bus; and (iv) *credit_i*, which is a control signal that indicates the buffer availability. In addition, the input port is composed of the following signals: (i) *clock_rx*; (ii) *rx*; (iii) *data_in*; and (iv) *credit_o*. These signals are the counterpart of the output port signals, respectively. Thus, each bidirectional link has six control signals and two data signals. Moreover, Lasio implements vertical links applying TSV technology. For instance, in Figure 12, Router 121 is connected to the Router 131 through a TSV bidirectional link.

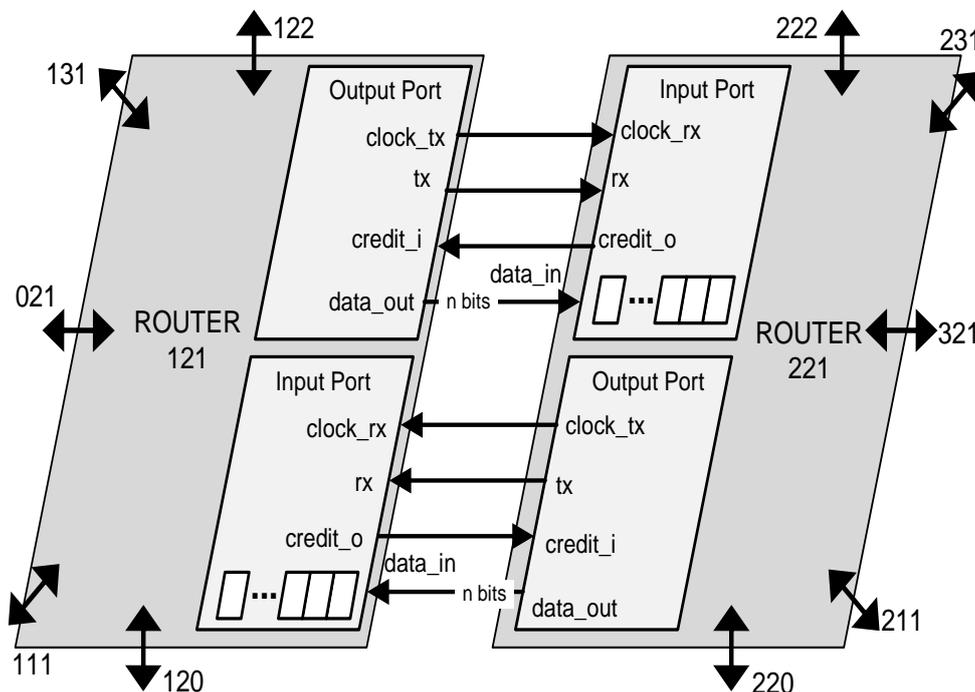


Figure 12. Example of Lasio signals between routers 121 and 221 [GHI12b].

3.3 Packet Structure

Figure 13 shows the packet structure employed on Lasio NoC, which is composed of an address field, a size field, and a payload field. The address field has the target router

address that is the XYZ coordinates of the target PE, the size field contains the quantity of flits in the payload and the payload encloses data of the application's messages.

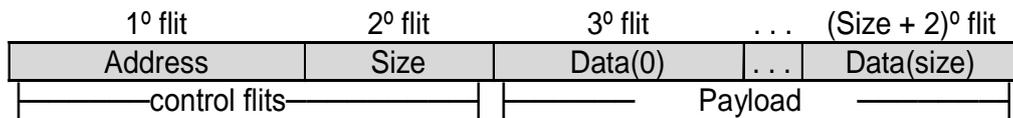


Figure 13. Packet structure of Lasio NoC [GHI12b].

3.4 Packet Routing

The routing algorithm states which path from a source node to a target node is used as routing path. This algorithm is classified as deterministic or adaptive. The former is employed when the path between the source and the target node is always the same; the latter is employed when the path between the source and the target node may change according to network parameters, such as traffic conditions.

Lasio implements XYZ deterministic routing algorithm, which is an extension of the XY routing algorithm explored in 2D NoCs. This routing algorithm is deadlock free and enables small area of implementation. When a router receives a header flit, the arbitration is executed, and if the incoming packet request is granted, the XYZ routing algorithm is performed to connect the input port data to the correct output port. From the source router to the target router, packets are routed firstly in X, after in Y and then in Z coordinates, respectively, passing through several buffers and ports. If the chosen output port is busy, the subsequent flits are blocked in the input buffer, and the request remains active until the connection with the port is established. When the target port is free, the arbitration algorithm takes place to decide which request will be served (in case of concurrent requests), establishing a connection between an input port and an output port.

There are three vectors called *in*, *out* and *available* located in a switching table. They are explored during the execution of the routing algorithm. The *available* vector is used to indicate the availability of a given output port, meaning if the port is transmitting a packet (i.e. busy) or if not (i.e. free). When there is a transmission request from an input port, the routing policy tries to find an available output port. In this case, all three vectors are updated. Otherwise, the input packet remains contained in the input buffer. The *available* vector position related to the output port is set as busy, while the *in* and *out*

vectors are interconnected. The *in* vector indicates to which output port the packet is being routed, while *out* vector indicates to which input port the packet is coming from.

Figure 14 and Table 2 exemplify the router switching process, where the *North* port has its output port set as busy because it is transmitting a packet from *West* input port. At the same time, the *North* input port is assigned to the *Top* port. When the packet transmission is finished, the *available* vector is updated (i.e. set to free).

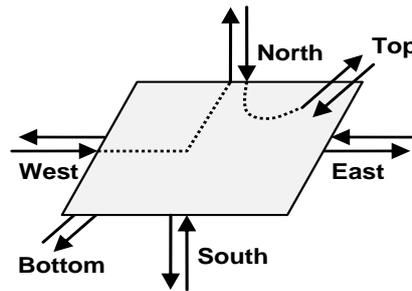


Figure 14. Example of two simultaneous connections in the router [GHI12b].

Table 2. Example of Lasio NoC switching table [GHI12b].

		Port name						
		East	West	North	South	Local	Bottom	Top
Vector	<i>available</i>	free	free	busy	free	free	free	busy
	<i>in</i>	-	North	Top	-	-	-	-
	<i>out</i>	-	-	West	-	-	-	North

3.5 Arbitration

3D NoC arbiter can be implemented in two ways: centralized or distributed. While in the centralized way, routing and arbitration mechanisms are implemented in a single module; in the distributed way, routing and arbitration are performed independently for each router port.

Centralized arbitration consists of a single module, which receives the buffers requests, and selects them to start packets transmission allowing the arbiter to receive concurrent and independently requests. This approach aims to minimize the resources of connection between the input and output ports considering all the packets that are ready to be transmitted from the buffers, as well as the current state of the output ports. The disadvantage of the centralized arbitration is that it imposes restrictions regarding packets routing capacity, which means that latency may be harmed since a single module

executes routing and arbitration. However, centralized arbiters are required in networks, which uses adaptive routing or input buffers with multiple queues [CUL98].

In the distributed arbitration, routing and arbitration are performed independently for each router port. Each input port has a routing module along with an arbitration module in its output port. Besides, this approach allows the construction of faster routers. Consider an adaptive routing algorithm, where a router can send requests simultaneously to more than one arbiter [CUL98]. If at least two arbiters answer the request, an output port must be chosen and the other one will be idle. This problem is solved if the arbiter has a global vision, such as centralized arbitration.

The Lasio centralized arbiter uses a dynamic rotating policy that prioritizes the packet routing on the input port using Round Robin algorithm. This method ensures that all incoming requests are processed, preventing starvation phenomenon. If the routing algorithm is able to establish a connection to the desired output port, another input port may require to the arbiter a new routing.

3.6 Switching

In a NoC, data are transmitted from a source to a destination via switches (or routers). To perform these transmissions, switches should assume a transferring data policy. The two switching policies normally used in NoCs are based in (i) *circuit switching* that establishes a complete path between the source node and the destination node before to transmit the entire message, or in (ii) *packet switching*, where a message is divided into packets and the source-destination connection is established during the packets transmission. The following sections present the packet switching method, since it is used in Lasio NoC.

3.6.1 PACKET SWITCHING

In cases where the messages transmitted between nodes are short and frequent, the circuit switching is unwarranted because it increases the network retention and the time to establish a circuit becomes much greater than the time in transferring messages. An alternative is to break the message into packets that are transmitted over the NoC.

The main advantage of this method is that the channel remains busy only while the packet is being transmitted. Its disadvantage is because each router requires buffers to store an entire packet or part of it, depending on the technique used. The three most

commonly used techniques for packet switching are: (i) store-and-forward (SAF); (ii) virtual-cut-through (VCT); and (iii) wormhole [RIJ01] [MOH98]. In this work, only wormhole technique is employed.

3.6.1.1 Wormhole

Dally and Seitz proposed wormhole-switching mode in [DAL86]. This variation of the VCT switching reduces the amount of buffers required to transmit packets on the network. In this method, the packet is divided into flits acting as a pipeline, where header flit (containing target node address) move across the network followed by other control and data flits. When this header flit is blocked, the remaining packet flits occupy the buffers of the intermediate switches (routers). Therefore, a packet must pass completely through a channel before releasing it to another packet.

An advantage of the wormhole if compared with VCT and SAF is the latency minimization. Another benefit is the reduction of buffers in intermediate switches, which do not need to store an entire packet, enabling the implementation of small and fast routers. Moreover, wormhole method does not allow deadlock in the communication. The wormhole switching is used frequently by offering more gains regarding network usage and routers cost.

3.7 Flow Control

The flow control mechanism determines the moment a packet should be transmitted to the next router. It serves to adjust the output rate of a source router with the input rate of a target router. Flow control is required whenever two or more packets demand the same resource simultaneously. When this occurs, one of the packets may be blocked, stored in a buffer or simply discarded [CUL98].

3.7.1 CREDIT-BASED

According to the definition above, Lasio uses credit-based flow control, which is an optimized communication mechanism, since it may consume few clock cycles to perform a flit transmission. This method utilizes FIFO buffers with customizable size at the receiver input, and a return line to the transmitter informing if there is space available in the buffer. The transmitter interprets this information as a credit; consequently, it just sends data if a credit is available. In credit-based protocol, the receiver sends to the transmitter via

credit_o/credit_i signals (Figure 12) an information indicating credits availability and the transmitter sends data only when there is credit available.

4 THROUGH SILICON VIA (TSV) INTERCONNECTION

A number of few enabling technologies - such as Package on package (PoP) - developed and introduced into the existing fabrication process flow have made 3D integration an interesting solution for SoCs development. Although, according to the 2007-ITRS roadmap [ITR07], the interconnections between stack dies could become one of the near-term (through 2015) “grand challenges” since both additional device and interconnection scaling could not fulfill the IC performance required. In this sense, 3D integration with TSVs, which was introduced by Savastiouk [SAV00] that wrote “Investment in technologies that provide both wafer-level vertical miniaturization (wafer thinning) and preparation for vertical integration (TSV) makes good sense, allowing to follow Moore's Law into the Z dimension”, aligned on a small pitch was one of the technologies identified to meet that challenge [PAP11]. Therefore, TSV might be seen as a key technology to help the semiconductor industry to extend the momentum of Moore's Law into the next decade [MAR12].

Consequently, in 3D integration technologies, the basic building element providing connections between the different stacked layers is the TSV-based vertical interconnection [VIV11]. They perform an important role for 3D ICs to not only establish efficient multicore computation systems, but also to reduce the core-to-core interconnection complexity and communication time. 3D improves the multicore operation performance allowing high device integration density and combining heterogeneous systems with good compatibility in the standard Complementary Metal-Oxide-Semiconductor (CMOS) process [BER07]. Moreover, the technology integrates more functionality into a smaller form factor with increased performance, lower power dissipation and reduced cost [GAR08].

Comparing vertical interconnections that employ TSVs technology in 3D NoCs to wire bonding and to metal bumps, TSVs are able to provide higher bandwidth. Hence, TSVs have become a major choice to replace them [BAN01] despite TSV interconnection has more complex fabrication method. Table 3 provides an overview comparing different technologies on 3D layers interconnections.

Table 3. 3D Layers Interconnections Technologies.

Technology	Advantages	Disadvantages
Wire bonding	High reliability; mature and cost effective processing	Low density; long wiring; large pad area
Metal bumps	Short length	Large solder ball
TSVs	Short length; high density; small footprint	Complex fabrication

4.1 TSV-Based 3D Integration Technologies

This section presents a short overview of technologies employed on 3D fabrication process. On top-level this process, three main techniques can be used to stack ICs:

- Wafer-to-Wafer (W2W) - complete wafers are stacked together and dies are extracted after assembling.
- Die-to-Wafer (D2W) - a die from one wafer is picked and placed on the top of another die integrated in a wafer.
- Die-to-Die (D2D) - mandatory if the assembly step is done by the manufacturer of the lower size die or by an external assembler.

The orientation of the individual dies in the stack is another item of variation. Dies can be connected Face-to-Face (F2F), Back-to-Back (B2B) or Face-to-Back (F2B) [BLA04], whereas face is the active layer and back is the passive silicon. In F2F stacking, the active sides of the two dies interconnect to each other. In this orientation, dies can be assembled without thinning step; however, F2F does not scale well to stacks of more than two dies [BLA04]. To interconnect dies using B2B stacking, TSVs in both dies are required. Besides, as F2F, B2B does not scale well to bulk more than two dies. F2B is the only technology explained here that allows stacking more than two dies and thus the only one compatible with our multiple dies stacking assumption.

Lastly, TSVs are usually made of copper or tungsten [MAR12], each one with different material properties. TSVs can be fabricated as via-first, via-middle or via-last, referring to their moment of creation in the processing flow, relative to front-end-of-line (FEOL) and back-end-of-line (BEOL) processing. TSV is via-first if they are processed before the front-end CMOS steps (transistors masks) [HEN07]. They are via-middle when they are processed after the front-end transistor process but before metal layer fabrication [PAR09]. Finally, via-last if they are implemented after the metallization stage [CHA08].

Typically, via-first and via-middle produce quite a lot smaller and denser TSVs than the via-last.

Figure 15 illustrates an expanded and general view of a 3D IC. It consists of 2D ICs that are thinned, bonded together and interconnected with TSVs distributed within the planes of the 2D ICs.

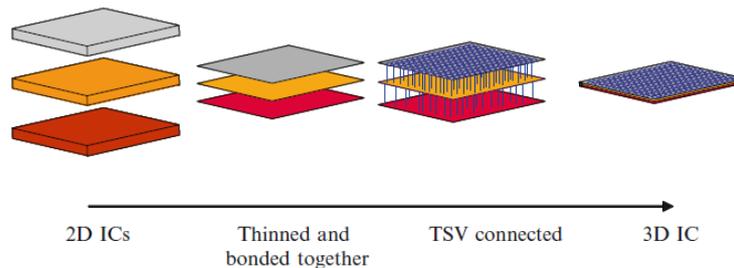


Figure 15. The basic structure of a 3D IC constructed from 2D ICs [PAP11].

TSV based 3D ICs contain pads on the wafer surfaces which are needed to bond to the vertical TSVs, using mechanical thermo-compression [PAS09]. Remark that pads are distributed on each layer and tend to be larger than via cross-section to account for the oxide coating. Thus, typically, 3D IC technology using TSVs faces some challenges such as dealing with parasitic capacitance that exists in TSVs as result of power consumed in vertical interconnections of 3D ICs [YON11], and with higher power and temperature densities. Such challenges raise punctual concerns:

- Firstly, as the number of TSVs used increases, the impact on area cost becomes more perceptible;
- Secondly, a large number of distributed TSV pads across the whole network may aggravate routing congestion [PAS09];
- Lastly, since current techniques of TSV fabrication still present a relatively low yield, more TSVs may result in lower product yield [LOI08].

The area cost might be seen as the main technical limitation of 3D integration, since the TSV pitch cannot probably go further beyond the μm scale [VIV11]. In this sense, Table 4 summarizes high-density TSV projections in order to forecast possible limitations on area. Remark that the TSV diameter is not considered the main restraint on area cost.

Table 4. High-density TSV projections in 2008 ITRS update [PAP11]

Principle parameters	Year							
	2008	2009	2010	2011	2012	2013	2014	2015
TSV diameter, D (μm)	1.6	1.5	1.4	1.3	1.3	1.2	1.2	1.0
TSV pitch, P (μm)	5.6	5.5	4.4	3.8	3.8	2.7	2.6	2.5
Pad spacing, S (μm)	1.0	1.0	1.0	0.5	0.5	0.5	0.5	0.5
Pad diameter, PD (μm)	4.6	4.5	3.4	3.3	3.3	2.2	2.1	2.0
Bonding accuracy, Δ (μm)	1.5	1.5	1.0	1.0	1.0	0.5	0.5	0.5

An interesting way to address the challenges mentioned above is over a TSV serialization scheme, which is implemented in the present work. It is an interesting alternative to reduce the TSV footprint on each layer enabling higher design space exploration, as well as more efficient core layout across multiple layers, mainly due to the reduced congestion. Moreover, fewer TSVs between layers would be cheaper and easier to manufacture.

4.2 TSV Serialization Scheme

TSV is considered a promising and efficient technology for 3D integration, since it displays compatibility with standard CMOS processes [BER07] and have been proved to be efficient in 3D integration [PAT06][BER07]. As Table 3 exemplifies, TSV is high-density, has low-capacitance interconnection, and hence allow more interconnections between stacked dies compared to traditional wire-bonds, and these interconnections can operate at higher speeds and lower power dissipation [MAR12].

However, TSV also brings some challenging problems that cannot be ignored, such as TSV requires the provision of pads in each layer [PAS09]. Relating the new TSV-based 3D die stacks and the semiconductor manufacturing, due to their many high-precision steps, both are defect-prone. Consequently, it is necessary to undergo electrical tests to weed out the defective parts and guarantee outgoing product quality to the customer [MAR12]. Besides, it is obvious that the maximum performance can be achieved by full layer-layer connection; however, as the number of tiles grow, it might not be practical to assume that each tile will be connected with corresponding TSV because of the limitation of manufacturing cost, chip area and the product yield as mentioned before. In addition, fewer TSVs between layers would be cheaper and easier to manufacture.

Therefore, a serialization scheme for communication on vertical links (i.e. TSVs) to minimize the above mentioned issues is proposed. Such scheme may be implemented at different levels depending on the employed routing algorithm and traffic scenario.

Moreover, TSV sharing potentially leads to substantial area minimization, enabling better design space exploration.

The serialization scheme proposed here is generic enough to be implemented at different layers levels depending on the employed routing algorithm and traffic scenario. Figure 16 shows the scheme proposed between two vertically adjacent layers to serialize communication between routers.

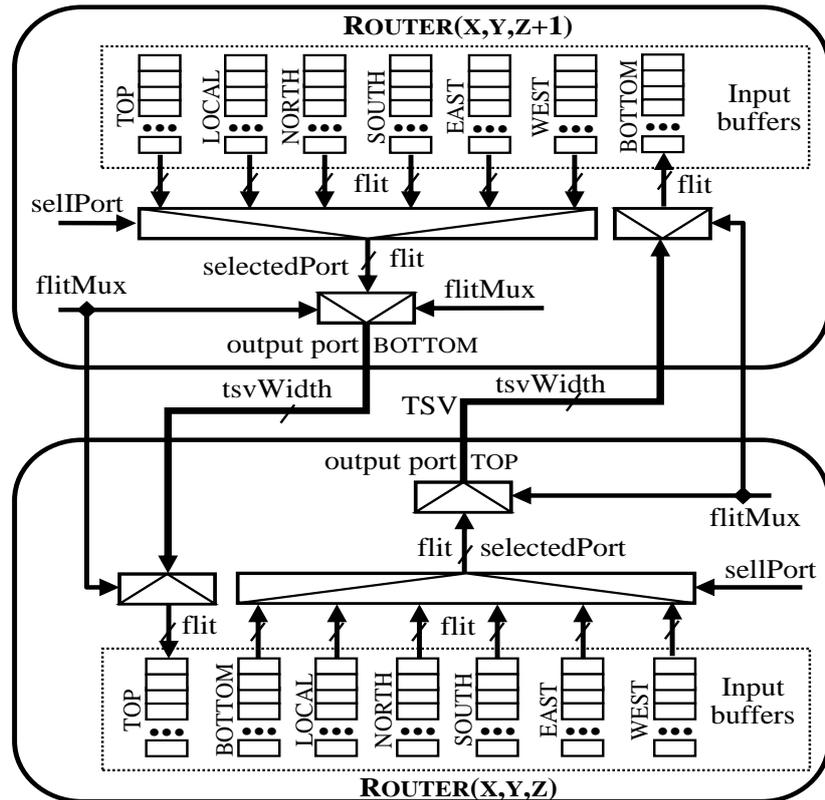


Figure 16. TSV serialization scheme.

TSV serialization occurs on the bottom-up direction when data is coming from router (X, Y, Z) to router $(X, Y, Z+1)$ and on the top-down direction, when data is to be sent from router (X, Y, Z) to router $(X, Y, Z-1)$, with $Z > 0$. For both directions, data comes from one of six input ports from the router port destination. For example, in case of bottom-up serialization, candidate input ports are NORTH, SOUTH, EAST, WEST, LOCAL and BOTTOM, whereas the TOP port should be the serialized data destination. Next, after the signal *sellPort* selects the input port, this data is directed to *selectedPort*, a flit-width bus. TSV serialization is controlled by *flitMux* - a signal with $\log_2(\text{flit width}/\text{tsvSize})$ bits - according to the number of TSV wires (i.e. *tsvSize*).

For instance, consider a NoC with 16-bit flits, 4-bit *tsvSize* and 2-bit *flitMux*, which serializes each flit in four steps. Once the signal is serialized, it is serially transmitted to the destination router through a TSV. At the target router, the demux circuit, which is controlled by the same signal *flitMux*, deserializes the data and transmits it to the corresponding input buffer, which is the TOP buffer in the top-down serialization scheme (or the BOTTOM buffer in case of bottom-up serialization). Hence, the number of serialized wires implies the number of steps (each requiring a single clock) needed to transmit all flit bits.

5 ELECTRA - 3D NOC GENERATION TOOL

According to ITRS assessment [ITR12], the construction of SoCs has become much more complex, especially due to the growing number of PEs. Such PEs are connected to routers that are further connected through a 2D or a 3D NoC. NoCs are an opportunity to improve SoCs overall performance, and since their complexity tends to increase, NoCs design requires new features such as support for different switching modes, routing algorithms, and message packets segmentation and reassembly procedures [OST05].

Hereupon, automatic and optimized tools for NoC generation and simulation have been proposed for several research groups as a way to relieve the NoC designers' effort. Examples of these tools are: (i) *NoCGen* [CHA04] creates 2D NoC descriptions files for posterior simulations and synthesis. It allows to select a set of templates varying routing algorithms and buffer depth; (ii) *MAIA* [OST05] is a framework for NoC generation and verification and it can produce Open Core Protocol (OCP) Network Interfaces (NI). Moreover, *MAIA* contains a module that verifies and creates data basic statistics regarding some time performance aspects; (iii) *BrownPepper* [BRU09] has a configuration interface to set necessary parameters for the network and traffic generation mostly according to user demands. It provides settings to run up to five simulations with different parameters. Remark that none of the three mentioned tools has support for 3D NoCs.

Independently of targeting 2D or 3D NoC, the fact is that automatic and optimized tools are important, in order to facilitate the integration of the network interconnections and the PEs guaranteeing fastness and accuracy since NoC-based architecture involves new design challenges, e.g., topology selection, router design, routing algorithm, communication protocols, system tools, and so on. All these challenges require a very time-consuming and error-prone design process of on-chip interconnections to design power-efficient and high-performance NoCs [OGR07]. Besides, such tools have to fulfill some requirements: (i) automated NoC generation; (ii) automated NoC-PEs interface and (iii) automated routers with their own crossbars, port numbers and buffers sizes. Furthermore, it is desirable that the tool may be able to produce different NoCs configurations for posterior network analysis, such as: (i) traffic patterns for different load conditions and source/target pairs, (ii) routing algorithms, (iii) injection rates, (iv) parameterizable network dimensions, (v) buffers depths, and (vi) packets sizes. With those parameters, it is possible to build NoCs according to requirements and boundaries of

some specific application or even generate different NoCs configurations for subsequent comparison.

Electra is a 3D NoC generation tool developed following all these previous requirements and aggregating a few more, which allows exploring and optimizing SoC designs employing the Lasio 3D NoC. Besides, using only the files that Electra produces, it is possible to simulate, validate, evaluate, and synthesize the NoC through commercial tools, such as *Modelsim* from *Mentor Graphics®* and other tools from *Cadence®*.

5.1 Electra Features

The design of NoC followed by simulation, functional validation and overall evaluations phases are essential to assessment of the correctness and performance of NoC and consequently of SoC architecture. Moreover, one pre-design step must be accomplished before the network design phase, which involves decisions regarding the communication paradigm choice, intellectual property core and infrastructure selection.

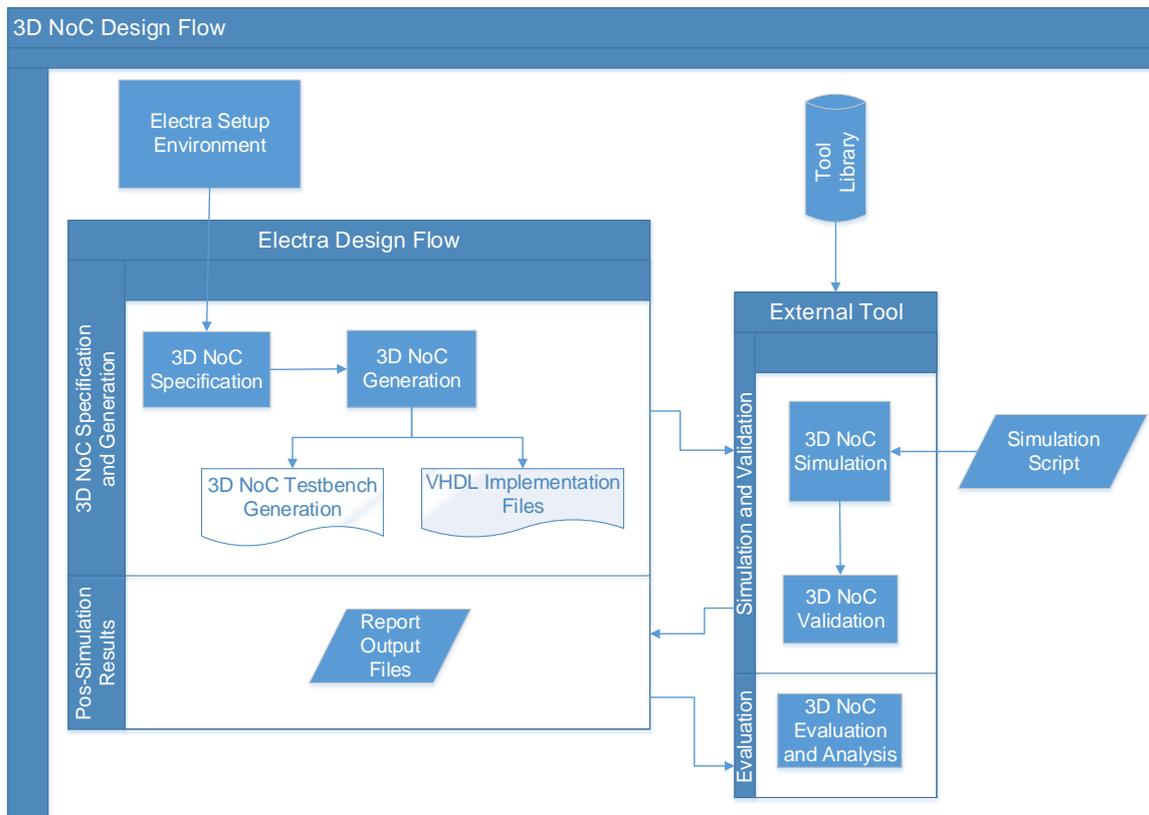


Figure 17. 3D NoC design flow.

The primary function of Electra is to accomplish the next step of the process - designing 3D NoCs with different and parameterizable configurations, which facilitates the

user task concerning NoC structure implementation and its internal functionality. The 3D NoC validation and evaluation steps are highly considered for future works; however, in the meantime, external tools are essential to complete these actions. Figure 17 illustrates the 3D NoC design flow followed by its description:

- *Electra Setup Environment* - Electra utilizes an environment responsible for automating processes regarding 3D NoCs architectures design flow;
- *3D NoC Specification and 3D NoC Generation* – comprise the selection of Lasio parameters (*3D NoC Specification*), e.g., network dimension, routing algorithm, synthetic traffic scenario, TSV serialization level, buffer depth and the generation of the 3D NoCs (*3D NoC Generation*). Then, a testbench is generated (*3D NoC Testbench Generation*) in order to instantiate the NoC and its parameters selected. Concurrently with this, *VHSIC Hard Description Language (VHDL) Implementation files* are produced containing all NoC description (detailed in Appendix A);
- *3D NoC Simulation and 3D NoC Validation* – it might be seen as one single step. An external tool, such as *Modelsim* is used in such phase. *3D NoC Simulation and 3D NoC Validation* step collects testbench generated and the VHDL output files to produce network data (*Report Output Files*);
- *Report Output Files* – The simulation of Lasio generates report files that contain several packets information (e.g. path traveled, time spent in each router, number of hops) to achieve some NoC measurements, such as latency and throughput, and also contain buffers information, like input and output time of a packet in order to assess the buffer occupancy. Such information is clock cycle accrued and it is included in the report output files (described in Appendix B). Such files are then used in the *3D NoC Evaluation and Analysis* step.

Figure 18 presents the NoC parameterization interface of Electra where it is possible to choose and set a wide range of parameters (despite the fact some values are not realistic, they can be considered for comparison purposes) to the network design, such as:

- Network dimensions (X, Y and Z);
- Flit width (8, 16, 32, 64 bits);

- Buffer depth (4, 8, 16, 32, 64, 128, 256, 512, 1024 positions);
- Routing algorithm (e.g. XYZ, bottom first);
- Packet size (8, 16, 32, 64, 128 flits);
- Injection rate (1%, 2%, 5%, 10%, 15%, 20%, 25%, 50%, 75%, 100%);
- Application size⁵ (e.g. 378, 3906, 7038 flits);
- Synthetic traffic scenarios (All-to-All, All-to-Bottom, All-to-Top, All-to-All-Complement, All-to-All-Next, Complement, Random);
- Number of traffics generated by each router – exclusively for Random traffic (1, 2, 4, 8, 16);
- TSV width (1, 2, 4, 8, 16 bits);
- The possibility to generate buffer information files, which are useful to measure the performance and the occupancy of the buffers.

However, a few parameters remain constant, which are issues of future work:

- Topology (mesh);
- Flow control (credit based protocol);
- Number of virtual channels by physical channel (1).

⁵ Represents the number of packets generated by each PE: $\text{Number_Of_Packets} = \text{Application_Size} / (\text{Packet_Size} - 2)$.

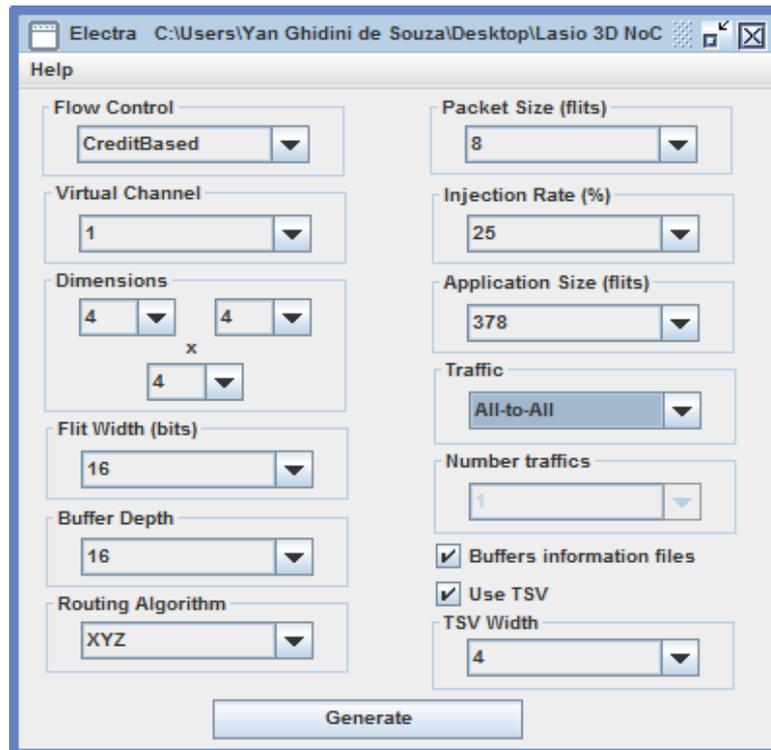


Figure 18. Electra 3D NoC generation tool.

All those parameters are incorporated to the VHDL implementation files containing the 3D NoC description. According to these predefined configurations, Electra is able to design 3D NoCs considering a specific application requirement or even design varied network configurations in order to compare their tradeoffs.

6 ENVIRONMENT SETUP

This chapter describes the setup used in all experiments. A general simulation environment was designed for latency, throughput, buffer occupancy, area, power evaluations and to measure the impact of serializing communication on vertical links of Lasio, as Figure 19 depicts.

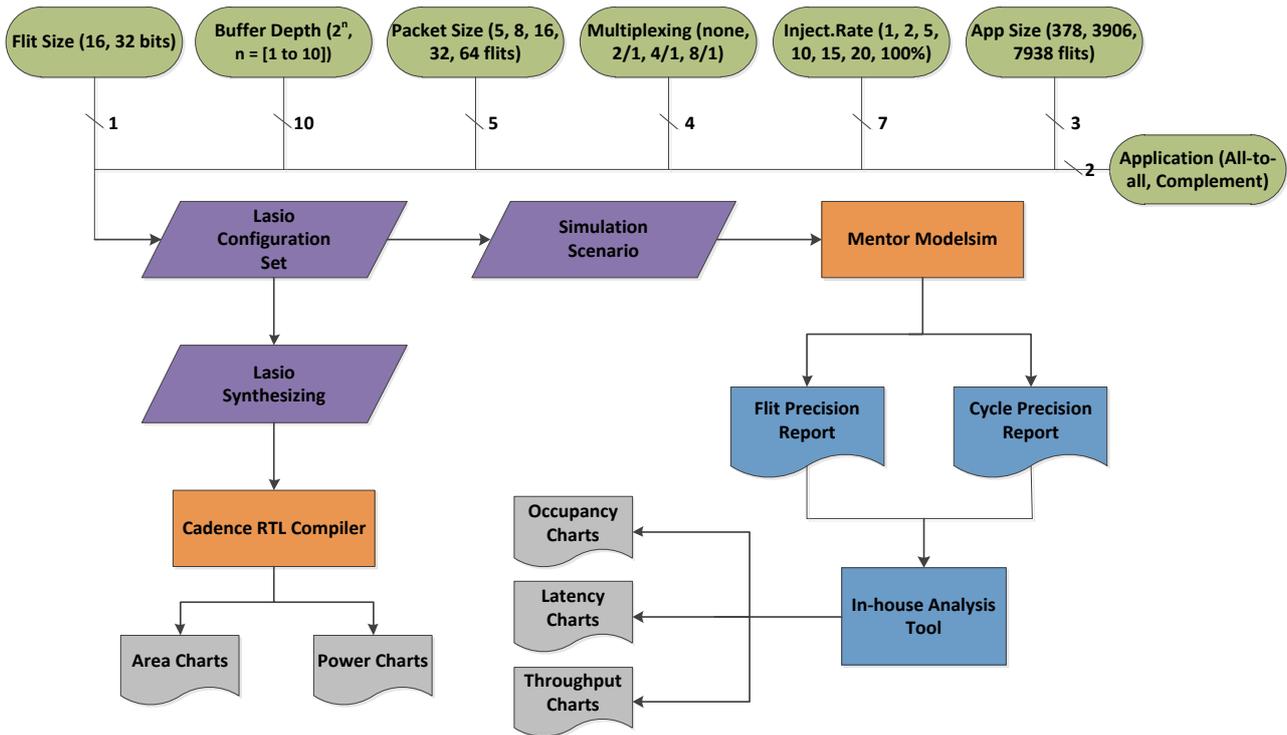


Figure 19. General environment setup.

Simulations consist of producer and consumer modules connected to each local port of each router in a 4x4x4 NoC Lasio, for instance. A consumer generates cycle accurate logs of the input buffer occupancy, of the latency and of the throughput for each transmitted flit, which allow the measure of these metrics. As regards area cost and power dissipation evaluations, Lasio was synthesized using Cadence RTL Compiler.

Since Lasio is parameterizable, variations in injection rate (which controls packet insertion into the NoC), packet size and input buffers depth allow identifying overall performance and the costs of serialization schemes for different network configurations. Moreover, various flit sizes, application sizes and serialization schemes might be employed during simulations, which contribute to more embracing NoC performance analysis.

The entire environment was described in VHDL and run with the Mentor Modelsim simulator. After simulation, an in-house tool have analyzed data generated by the consumer and then produced charts representing (i) average and worst case of occupancy of router input buffers, (ii) average, worst and best case of network latency, (iii) total application latency, (iv) average network throughput and (v) average application throughput. In order to analyze (vi) area footprint and (vii) power dissipation, Lasio routers were synthesized targeting the 65 nm STMicroelectronics CMOS technology.

All the assessed metrics are strongly dependent on the communication pattern. However, when choosing the synthetic traffic scenarios below (*All-to-All* and *Complement*) whose characteristics are determinism and uniformity of packet load, evaluations of latency, throughput, occupancy and serialization impact might be made independent of the communication pattern.

6.1 Traffic Scenarios

The following different synthetic traffic scenarios were adopted as producers: (i) *All-to-All* and (ii) *Complement*. The purpose of choosing synthetic scenarios is cover a wide variety of traffic situations, as depicted in Figure 19, that stress the use of NoC resources without concerning PEs mapping aspects. It is acknowledged that more realistic traffic scenarios could extend and contribute even more with the analysis, which might be considered for future works.

6.1.1 ALL-TO-ALL

In this traffic scenario, all nodes send the same amount of data (i.e. uniform packet load) with a pre-determined injection rate and in a deterministic way to all others nodes, except to itself. Firstly, all nodes simultaneously send a packet to node 0, and then all nodes send packets to node 1, and so on. This traffic model covers several traffic and blocking conditions, forcing that a large number of packets traverse the network simultaneously. However, *All-to-All* traffic scenario is not the best approach when dealing with real applications, since the communication destinations change frequently.

6.1.2 COMPLEMENT

Complement traffic scenario generates and injects packets into the network in a similar way; however, in this scenario each router sends packets to the router with

complementary address. The complement function associates each router identifier to another router identifier, located in some way, in an opposite NoC position (e.g. in a 3x3x3 NoC, the complement of address 000 is address 222, the complement of address 120 is 102, and the central router of address 111 does not have complementary address). The choice for the *Complement* scenario derives specially from the fact that it uses vertical interconnections for all flit transmissions causing a vertical communication “stress”, and consequently, allowing an efficient analysis of the proposed serialization scheme.

6.1.3 TRAFFIC SCENARIOS VARIATIONS

Despite the fact Electra supports the following varied traffic scenarios generation (*All-to-All-Next*, *All-to-All-Complement*, *All-to-Top*, *All-to-Bottom* and *Random*) and they present interesting characteristics for further network evaluations allowing a thorough analysis of 3D NoCs, they are not used in the context of this work. Nevertheless, since they are considered for future works, the next subsections briefly describe these scenarios.

6.1.3.1 All-to-All-Next

This traffic scenario is a variation from *All-to-All* scenario. Here, each node (*source_address*) begins sending a packet to its right neighbor (*target_address*), whereas for each dispatch the *target_address* is incremented in one position inside the network until it reaches the *source_address* left neighbor completing a round in the network. In other words, *All-to-All-Next* performs a circular transmission inside the NoC. The idea of using this traffic scenario is to diminish the output data congestion, when compared to pure *All-to-All* traffic that implies a hotspot traffic in the target PE.

6.1.3.2 All-to-All-Complement

All-to-All-Complement is also a variation from *All-to-All* traffic scenario. Besides, its behavior follows the *All-to-All-Next* logic with one simple difference: each node (*source_address*) begins transmitting to its complementary position node (*target_address*), then to its complement neighbor node and so on, until the transmission completes a turning. *All-to-All-Complement* performs a different traffic distribution enabling to explore some other congestion scenarios, especially in terms of high data transmission in vertical links.

6.1.3.3 All-to-Bottom and All-to-Top

All-to-Bottom and *All-to-Top* are traffics designed to use vertical links as first option of sending packets between routers. The idea of both scenarios is over utilizing the vertical channels. Possibly, either *All-to-Bottom* or *All-to-Top* combined with the TSV serialization scheme will improve overall communication performance, since vertical communication channels are typically underused [GHI13a]. Regarding *All-to-Bottom*, the target routers will all be located in the first NoC layer, and all source routers (located in the above NoC layers) will send packets to those targets. *All-to-Top* follows the same method, except that the target routers will be located in the last layer and source routers in the layers below.

6.1.3.4 Random

In this scenario, all routers send packets to all other target routers (except to itself) in a randomized manner. Such random targets are defined in execution time. *Random* traffic does not operate with uniform packet load enabling to determine possible congestion areas in a 3D NoC communication pattern due to its traffic unpredictability, indicating for instance, where are the best places to position a TSV, how many TSVs are necessary, and the level of its serialization.

6.2 Metrics Evaluated

6.2.1 PACKET LATENCY

Figure 20 shows that packets latency metric is observed in different ways. The communication latencies presented here section are not limited to the packets delay into the NoC, they consider the packets that are delayed to enter into the NoC as well. Next, we introduce some reasons to make the differentiation of the transmission latencies according to the distribution of packets injection and packets reception easier.

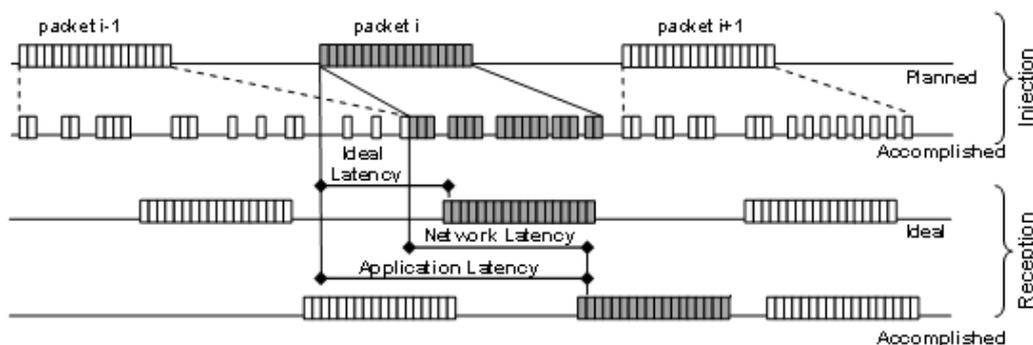


Figure 20. Communication latency metrics [MOR10].

The **planned injection** is the moment that a packet can be injected into the NoC. In those simulation scenarios, all packets are specified in an input text file with its planned injection times. **Accomplished injection** considers the exact insertion timing of a packet into the NoC, which may be different from that defined in planned injection, due to the occurrence of contentions. The **ideal reception** represents the estimated time of packets delivery. The **accomplished reception** shows the real delivery time of packets at their destinations. Figure 20 [MOR10] shows distributions of such injection and reception scenarios. The **ideal latency** is the minimum number of clock cycles that a packet needs to reach its destination [MOR10]. This value is obtained from the difference between the planned injection time of the packet and the expected delivery time of the same packet.

The concept of **NoC latency** here is related to the transmission delay of a packet from source to destination, which can be influenced by other packets competition for NoC resources. On the other hand, **application latency**, or App latency, expresses the time spent between the moment a packet is created by the application and the moment the packet is consumed by the target node. App latency illustrates the most important impact on the ideal performance of a communication, since it is computed as the difference between the planned injection time of packets and their exact delivery moment at destination [MOR10].

6.2.2 PACKET THROUGHPUT

Another parameter considered to evaluate NoCs performance is the packet throughput. Similarly as latency, both **NoC throughput** and **application throughput** (App throughput) are considered. While NoC throughput means the NoC capacity on packets transmission in a given interval of time (i.e., the average number of flits arriving in all PEs per clock cycle), App throughput considers the packets delay before entering on network, which starts since the moment that a packet is available to be transmitted. Buffer depth, traffic scenario, packet size and other NoC resources competitions might influence both throughput metrics.

6.2.3 BUFFER OCCUPANCY

Buffer occupancy on router input ports connected to TSVs display the average and the highest occupancy of top and bottom ports buffers of the NoC. The reason to evaluate vertical communication channels is to analyze the impact of TSV serializing scheme on the buffer occupancy.

6.2.4 AREA CONSUMPTION AND POWER DISSIPATION

Area consumption and power dissipation are critical issues in 3D ICs. Although the total power dissipation of 3D systems is expected to be lower than that of mainstream 2D ICs [PAV08] – since the global interconnections are shorter – the increased power density along with greater area footprint become significant challenges in the 3D paradigm. In order to evaluate area consumption and power dissipation, a central router Lasio, which contains all seven ports, was synthesized targeting the 65 nm STMicroelectronics CMOS technology. Additionally, the implementations were designed for operating at 1 GHz.

7 LASIO PERFORMANCE EVALUATION

This chapter presents an extensive architectural exploration on the Lasio 3D NoC. It analyzes latency and throughput, for both network and application, buffer occupancy, area consumption, and power dissipation as performance evaluation metrics, varying NoC parameters, such as injection rate, buffer depth, packet size, application size and specially modifying the design with and without serialization of TSVs. All experiments presented assume that Lasio architecture contains 64 tiles and routers in a cubic format, 4x4x4 mesh. However, we acknowledge that, in practice, 3D mesh networks are not expected to be symmetric in 3D chip designs. The number of device layers is expected to be much less than the number of processor tiles that can be placed along the edge of one layer [RAM09].

Next sections present a Lasio architecture evaluation considering metrics recently mentioned. Section 7.1 contains a comparison between 2D and 3D NoC topologies and it contemplates Lasio architecture results considering that interlayer communication have the same cost as intra-layer communication; Section 7.2 encompasses a 3D NoC study that indicates the possibility of applying serialization scheme on TSVs; and in Section 7.3, results consider and analyze different serialization levels of the TSVs.

7.1 2D and 3D NoC Topologies Comparison

This section analyzes the architectural impact of 2D and 3D NoC topologies on network and application latencies and on data throughput. These evaluations also show the significant impact of buffer depth on application latency for 2D and 3D topologies, which cannot be neglected in NoC design exploration. Such results are available and can be seen in two different publications – [GHI12a] and [GHI12b].

7.1.1 NETWORK AND APPLICATION LATENCIES

In network and application latencies, *All-to-All* and *Complement* traffic scenarios are used with injection rate of 800 Mbps, which in these experiments means 100% of injection rate. The buffer depth is set to 2^n flits, where n ranges from 2 to 10. To provide a rational comparison Hermes 2D NoC [MOR04] is an 8x8 mesh (64 tiles and routers), does not contain virtual channels and it has credit based control flow (the same as Lasio). Packet

size varied in some experiments from 5-flit packets to 1024-flit packets (considering each flit is 16 bits width).

The first approach compares the performance of Lasio 3D NoC to its 2D counterpart. Both NoCs were initially experimented with 5-flit packets and *All-to-All* traffic pattern.

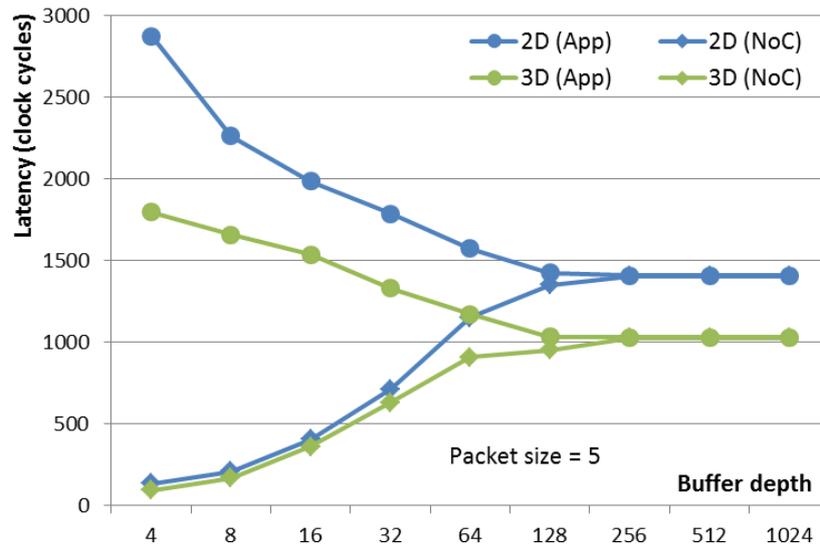


Figure 21. NoC and App latencies comparison between Lasio 3D NoC (4x4x4 mesh) and Hermes 2D NoC (8x8 mesh).

The first noteworthy aspect in Figure 21 is the minimization of the packets latency on the Lasio three-dimensional NoC compared to its 2D counterpart for both, application and network latencies, and for any buffer depth simulated. Besides, when comparing 2D and 3D NoCs performance, one should analyze the tradeoff existent between buffer depth and topology. Observing Figure 21, for 128-flit buffer depth or higher, the difference between App latency and NoC latency tends to zero in both 2D and 3D NoCs. This behavior indicates that the latencies are no longer influenced by buffer depth, thus the use of bigger buffers is not necessary for transmitting 5-flit packets. Moreover, the preliminary results highlight that when applying an appropriate buffer depth, the App latency is reduced, e.g., in these experiments up to 3.4 times for 2D topologies and 2.3 times for 3D topologies.

Figure 22 shows the behavior of the NoC latency and the App latency of a 3D network on chip under *All-to-All* traffic pattern, for different buffer depth, and k-flit packet sizes (k varies from 5 to 64). It is a fact that the maximum performance is achieved when App latency is equal to NoC latency, since in this case NoC promptly consumes packets

that are injected by the application. For instance, 32-flit packet size has as optimum a 512-flit buffer depth.

Observing Figure 22, it is noticeable that greater is the packet size, greater is the observed App latency, the same behavior is not observed for NoC latency, since it is more dependent on buffer depth and packet size relation. This last behavior may be justified by the influence of routing and switching strategies. In both comparisons, Figure 21 and Figure 22, the increase of buffer depth implies in a gradual slight decrease of App latency; however, it compromises NoC latency. This phenomenon can be explained by the fact that the increase of buffers enables better distribution of packets into the NoC reducing contention and, on average, approximating the payload flits to the target routers.

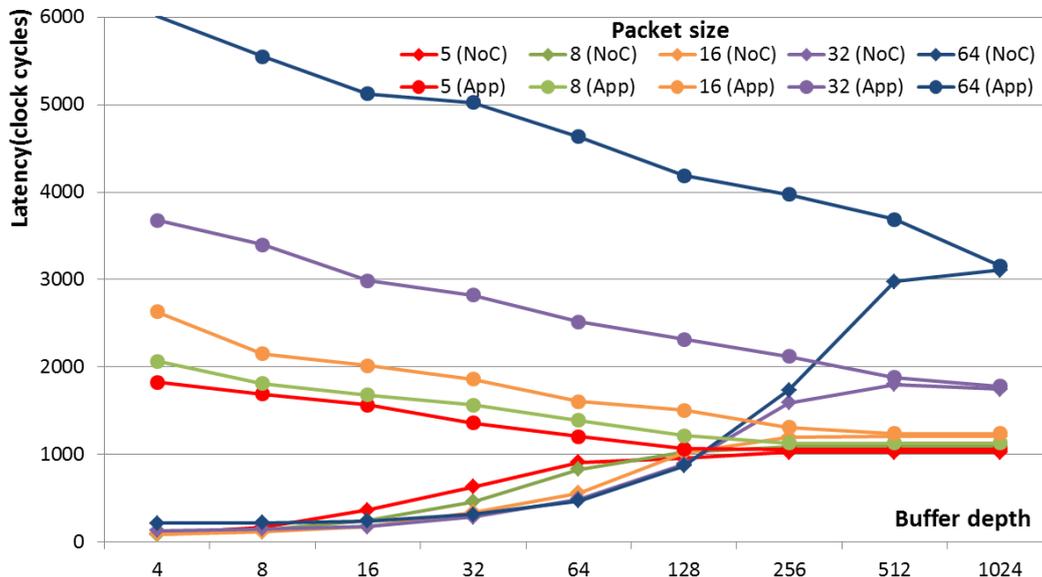


Figure 22. NoC latency versus App latency for different buffer depth and packet sizes.

Figure 23 displays the traffic influence in NoC latency and in App latency according to different buffer depth. In order to evaluate both latencies, an 8-flit packet size was utilized.

Figure 23 shows that, for all buffer depth, both NoC latency and App latency of *Complement* traffic are greater than the ones observed in *All-to-All* traffic scenarios. The greater number of hops performed by *Complement* traffic pattern can explain this behavior. In addition, *Complement* traffic implies the same NoC latency and App latency curves behavior pointed in Figure 21 and Figure 22, i.e. the increase of buffer depth decreases App latency and increases NoC latency until a given buffer depth.

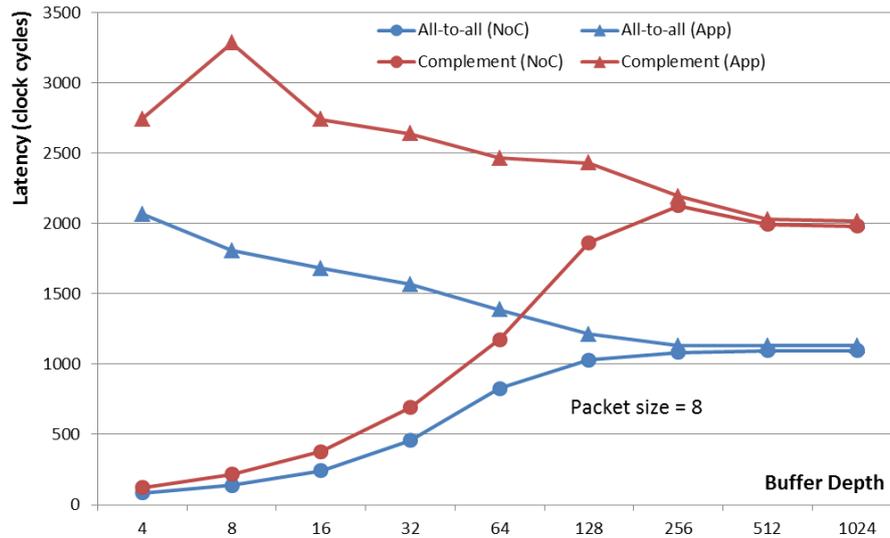


Figure 23. Traffic influence on NoC latency and on App latency.

7.1.2 NETWORK AND APPLICATION THROUGHPUTS

As latencies results, in network and application throughputs, *All-to-All* and *Complement* traffic scenarios were used with injection rate of 800 Mbps (i.e. 100% of injection rate) and buffer depth set to 2^n flits, where n ranges from 2 to 10. Here, Hermes 2D NoC and Lasio 3D NoC have the same architecture detailed in Section 7.1.1.

The throughput of the communication infrastructure generally depends on the traffic pattern, but not only. In these experiments, other dimensions were studied such as the relation of buffer depth and packet size under *All-to-All* traffic pattern.

Figure 24 shows that the NoC throughput is directly dependent on the packet size applied, i.e. the NoC throughput is increasing as the packet size augments. This behavior is independent of the buffer depth. However, smaller buffer depth has guaranteed the highest NoC throughput in these experiments. On the other hand, the App throughput remained almost constant for any buffer depth and packet size.

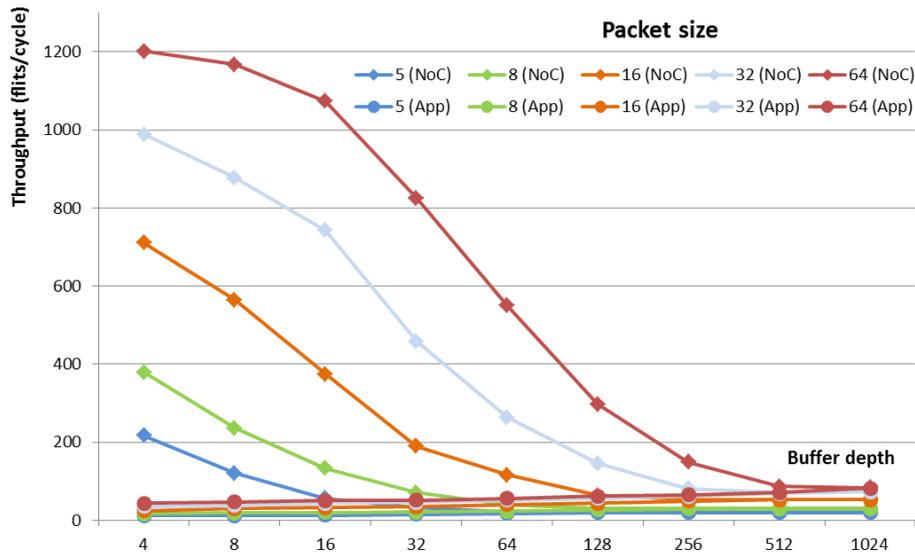


Figure 24. NoC throughput and App throughput behavior according to five sizes of packets and nine depths of buffer.

Figure 25 illustrates the traffic influence on both NoC and App throughput according to nine depths of buffer, considering an 8-flit packet size.

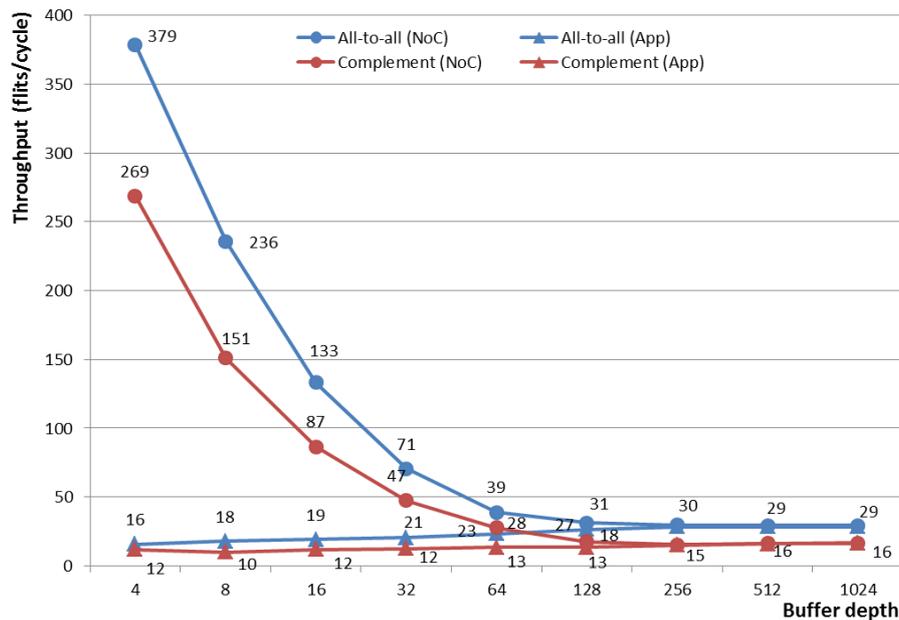


Figure 25. Traffic influence on NoC and on App throughput.

Observing the Figure 25, NoC throughput is superior to the App throughput in both traffic scenarios regardless buffer depth until 256-flit, when the NoC and App throughput values are approximately the same for each traffic pattern. Moreover, *All-to-All* traffic pattern presents higher throughput than *Complement* scenario. Examining the results, it has noticed higher NoC throughput for both traffic scenarios when smaller buffers are

utilized. On the other hand, the App throughput increases according to the buffer depth, and it stabilizes from 512-flit buffer for both traffic scenarios. This set of experiments enables to detect more appropriate buffer depth in order to attend some 3D NoC design requirement.

7.2 Buffer Occupancy Analysis and the Serialization Technique on the 3D NoC Lasio

This section analyzes buffer occupancy on the Lasio and the possibility of TSV serialization without performance degradation at system level. Such analysis might be seen at [GHI13a].

7.2.1 BUFFER OCCUPANCY

The parameters indicated below were used during the Lasio performance analysis and evaluation regarding buffer occupancy.

- Injection rate: packets injection rates of 100% of the maximum channels capacity was implemented and evaluated. This percentage can be translated as 800 Mbps injection rate;
- Buffer depth: 4-flit positions;
- Synthetic traffic scenario: All-to-All;
- Packet size: The size used in the experiment was 8-flits (considering 16-bits width for each flit);
- Application size: Electra enables to define the application size applied to Lasio and it undertakes of determining how many packets per router are required to transmit the whole application. This number is given as $(App / (Pk - 2))$, where App and Pk are the application size and the packet size, respectively. The experiment realized considered an application size equals to 4032 flits;

Figure 26 shows the measured average and highest buffer occupancy of top ports (i.e. ports connected through a TSV to a highest level of the 3D NoC) for the study case. The results were obtained by measuring the average percentage occupancy for all routers of the NoC. As the charts show, higher layers present typically higher occupancy levels. One of the reasons for that is the fact that these layers will generally present higher congestion traffic scenarios, given the XYZ algorithm. However, more importantly, the

obtained results suggest that vertical communication channels are typically underused. In the worst case, buffers occupancy reached a peak of 31% and an average of less than 22%. Given the 100% injection rate, and the high congestion *All-to-All* scenario, these values represent upper bounds for the usage of top ports. In this way, schemes to serialize the TSVs required by such ports are enabled.

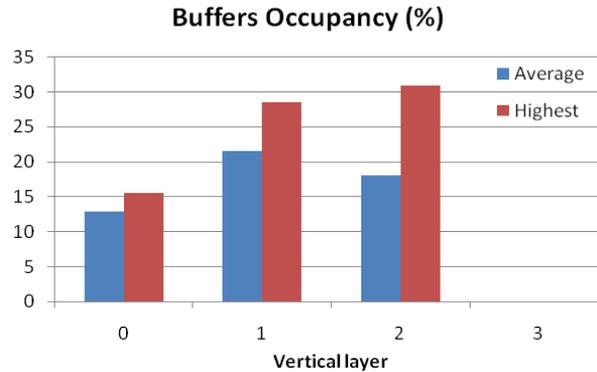


Figure 26. Top ports buffer occupancy.

TSV sharing potentially leads to substantial area and power improvements. For instance, given the worst-case peak of 31% of occupancy, the top TSVs of the network could employ 3/1 serialization (i.e. 3 communication channels sharing 1 physical link). In addition, preliminary results, application mapping and packet, flit and buffers dimensioning can obtain points that TSV serialization will jeopardize network latency and further optimizations. These results also display that similar occupancy levels are obtained for bottom ports buffers. In this way, it is presumable that all interlayer links of the NoC can take advantage of employing TSV serialization.

A deeper analysis was performed on the distribution of the occupancy displayed in Figure 26. Figure 27 (a), (b) and (c) displays the occupancy of top ports buffer for each router of the three lower layers, 0, 1 and 2, respectively. Results for the top layer are omitted because this layer has no connection to the top. As the tables suggest, further optimizations can be obtained by exploring serialized TSVs partitioning. For instance, given the sets of routers $A = \{00, 01, 11\}$ and $B = \{22, 23, 32, 33\}$, set A typically presents higher peaks of occupancy. Its peak is of almost 31%, while the peak for set B is of less than 25%. In this way, set A should employ 3/1 TSVs serialization, while set B could be more relaxed and employ 4/1. In addition, these optimizations can be explored at the different layers.

YX	(a)				(b)				(c)			
	0	1	2	3	0	1	2	3	0	1	2	3
0	14.7	13.3	12.0	11.1	28.6	20.5	20.6	18.5	30.9	16.4	16.6	12.7
1	15.5	14.1	14.6	12.1	26.5	25.3	24.1	19.3	23.6	23.5	24.2	16.7
2	13.7	13.8	12.9	12.7	23.9	21.9	24.2	19.0	18.0	18.3	20.7	14.8
3	11.3	11.2	12.3	10.7	19.0	19.0	19.6	15.9	12.3	14.2	14.7	10.9

Figure 27. Buffer occupancy of top ports (in percentage) for each router of the three lower layers 0 (a), 1 (b) and 2 (c).

7.3 Serialization Effect on the Lasio 3D NoC

This section evaluates a TSV serialization mechanism. It provides an assessment of network and application latency, as well as input ports buffers occupancy. Moreover, it explores area consumption and power dissipation on the Lasio with different serialization degrees. These results might be seen at [GHI13b].

7.3.1 LATENCIES AND BUFFER OCCUPANCY

On the next simulations, different injection rates were useful for evaluating the impact of the proposed serialization scheme under different levels of traffic congestion. The simulations employed injection rates of 1%, 2%, 5%, 10%, 15% and 20% for each scenario. These injection rates allow realistic comparisons, as many references point that real applications most often inject traffic in NoCs at rates below 15% [DAL04]. Flit size was assumed to be 16 bits and application size was pre-determined to be 378 flits. Albeit other flit sizes and application sizes were also simulated, results varied only quantitatively and not qualitatively. Thus, these results were omitted here. Simulations used five different buffer sizes: 4, 8, 16, 32 and 64. Packet sizes varied among 8, 16, 32 and 64 flits, and four serialization schemes were evaluated: 1/1, 2/1, 4/1 and 8/1. In the 1/1 serialization scheme, it is assumed the availability of one TSV for each bit of top and bottom ports, in other words this is equivalent to an scheme without serialization, which is applied for comparison purposes. In 2/1, 4/1 and 8/1 serialization schemes, on the other hand, it is assumed one TSV for each 2 bits, for each 4 bits and for each 8 bits, respectively. Eighty different configurations of Lasio was reached by varying each parameter at a time (i.e. 5 sizes of buffer \times 4 packet sizes \times 4 serialization schemes). Additionally, applying six injection rates with two types of traffic on the 80 Lasio configurations generates 960 different simulations.

Figure 28, Figure 29 and Figure 30 present only results for the *Complement* traffic scenario regarding average latency results, when varying the serialization level as a function of flit size, injection rate and buffer size. The first column of charts represents the *measured latency* (in clock cycles) and the second column represents *relative gain* (in percentage). This relative gain was measured as the expected losses using the serialization scheme divided by the measured latency overhead. For instance, assuming an 8/1 serialization scheme (MUX 8/1), it is expected an eight times bigger average latency, which is divided by the measured latency. In this way, it is possible to evaluate the tradeoffs of using serialization schemes.

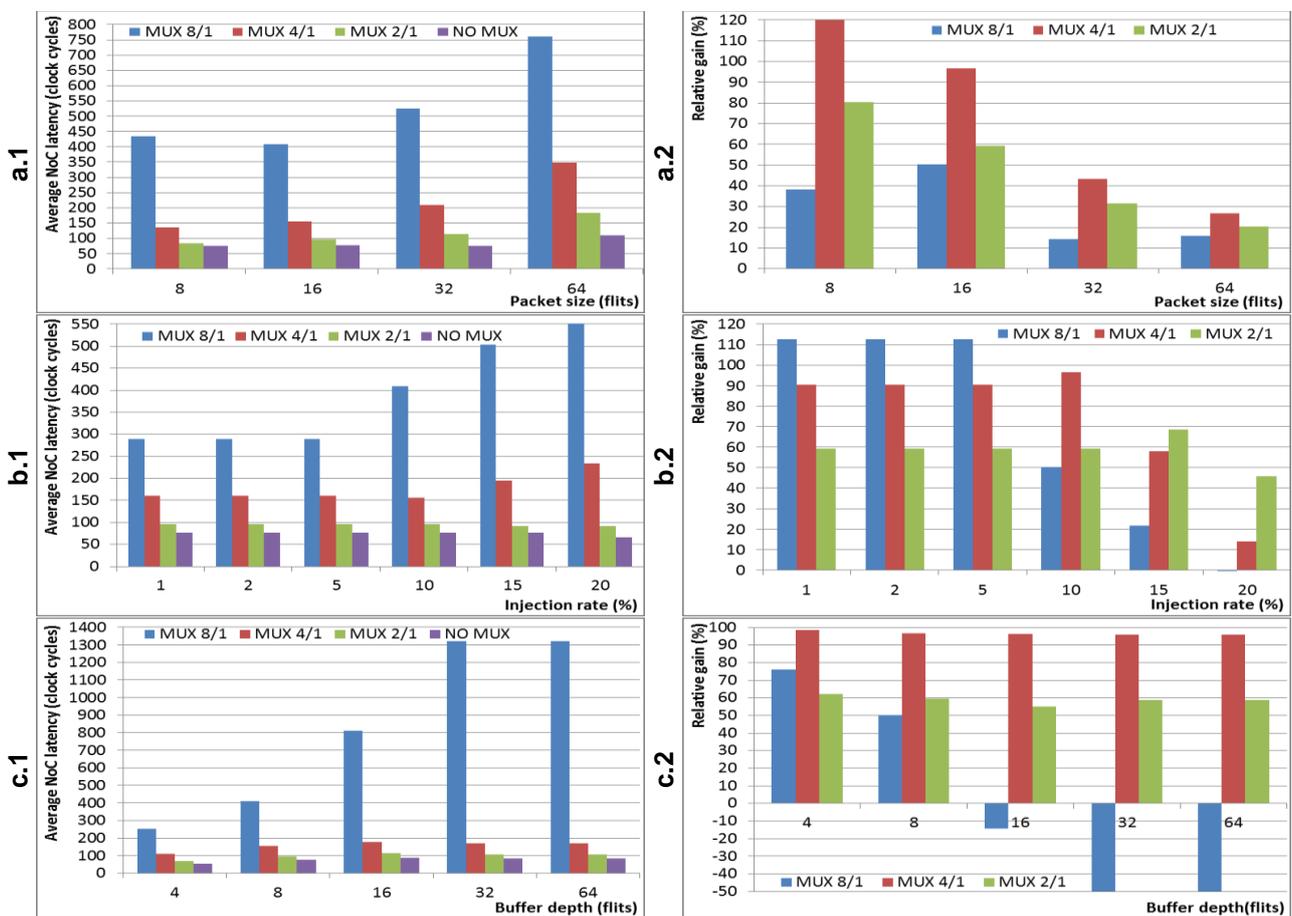
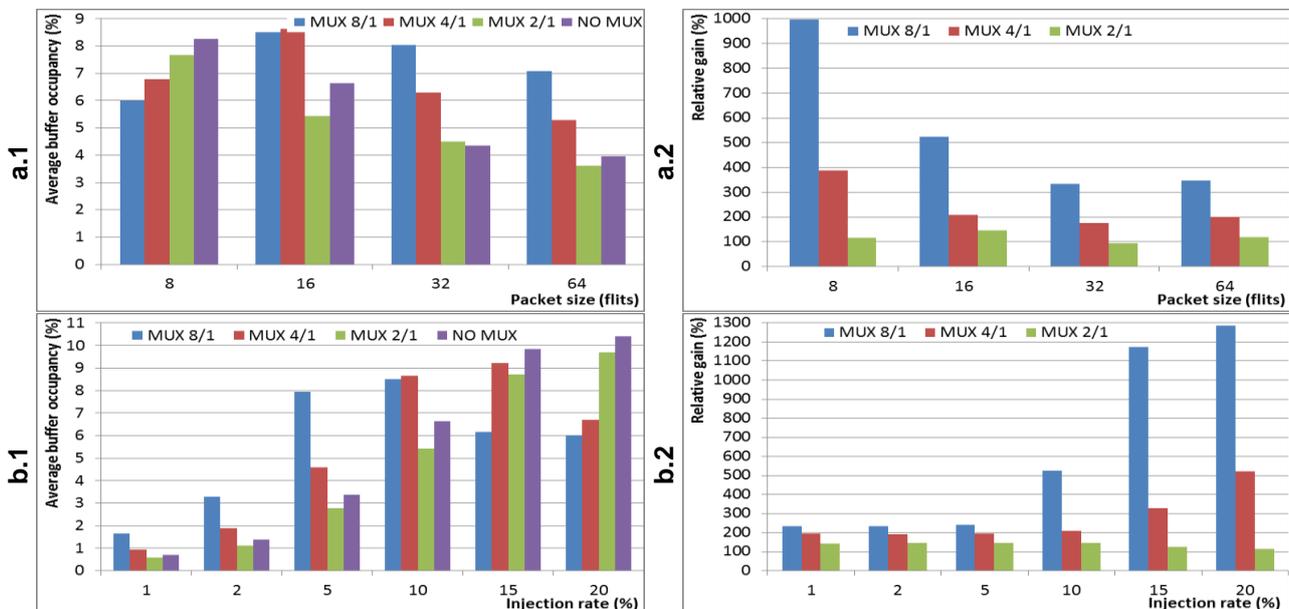


Figure 28. Average NoC latency (first column) and relative gains (second column) for the following variations: (a) packet size (10% of injection rate and 8-flit buffer depth); (b) injection rate (16-flit packet size and 8-flit buffer depth); (c) buffer depth (10% of injection rate and 16-flit packet size).

Figure 28(a) shows the impact of the packet size in the average network latency. The presented results are for a fixed injection rate of 10% and 8-flit buffers. Figure 28(b) shows the impact of the injection rate, with fixed packet size (i.e. 16 flits) and 8-flit buffers,

and Figure 28(c) shows the effect of buffers depth, for a fixed injection rate of 10% and 16-flits packets. As the first column charts of Figure 28 show, the bigger the injection rate, the packet size or the buffers depth, the bigger is the average network latency. This is valid for all scenarios, independent of the serialization scheme.

Figure 28(a) shows that for all packet sizes albeit MUX 8/1 Lasio requires 8 times less TSVs than an equivalent NoC without serialization. It does not increase 8 times the average network latency; In fact, the serialization scheme slightly increases the network latency (e.g., for 8-flit packet size, the NoC latency increases less than 4.8 times). Therefore, we can assume a relative gain of roughly 38% as showed in Figure 28(a.2). Accordingly, all charts of the second column of Figure 28, i.e. Figure 28(a.2), Figure 28(b.2) and Figure 28(c.2), present relative gains. These values are useful to measure the cost-benefit relationship of the proposed scheme. As charts show, MUX 2/1 and MUX 4/1 always present positive relative gains, independent of the scenario, which indicates that the serialization scheme can help coping with technological problems without worsening average network latency. On the other hand, the MUX 8/1 presented negative gains for big buffers and for big injection rates, which suggests the lack of scalability of this scheme. In this way, it is possible to identify a saturation point for TSV serialization on the Lasio NoC. Similar results are expectable for other NoC architectures.



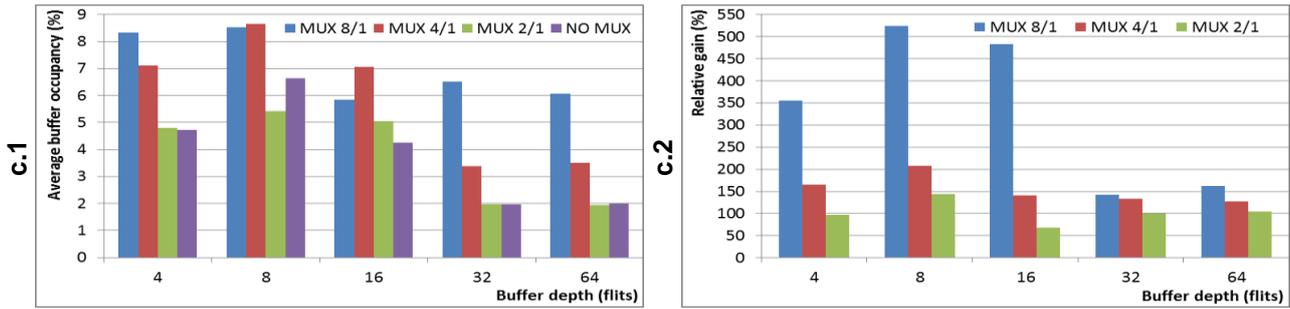
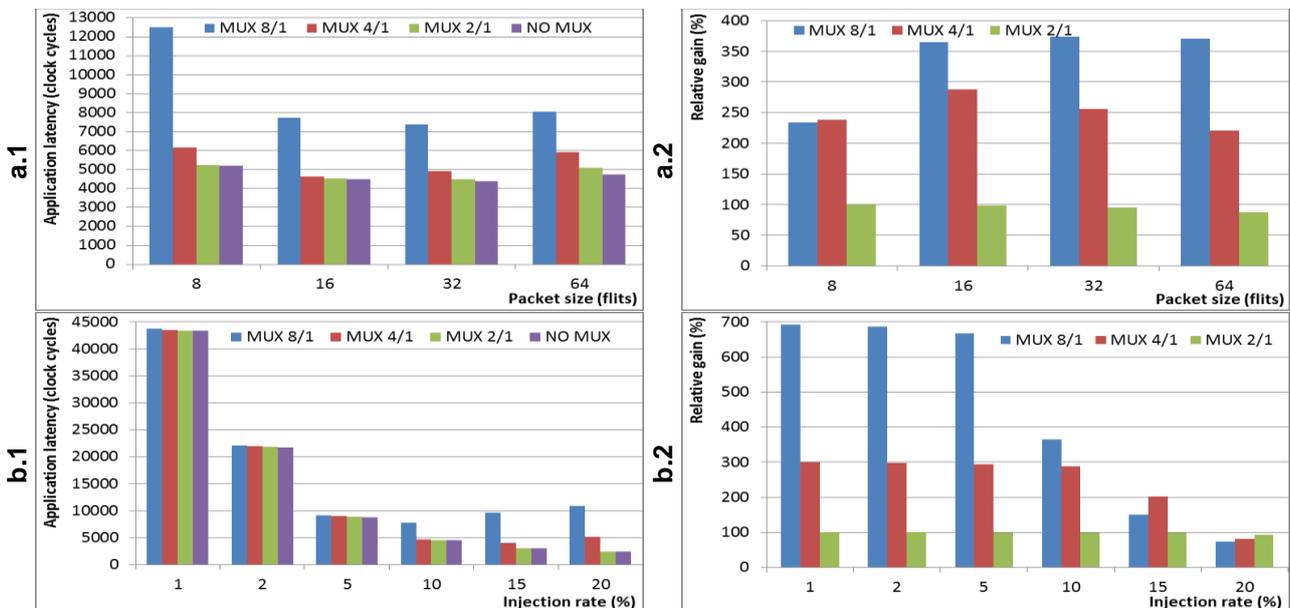


Figure 29. Average buffers occupancy (first column) and relative gains (second column) for the following variations: (a) packet size (10% of injection rate and 8-flit buffer depth); (b) injection rate (16-flit packet size and 8-flit buffer depth); (c) buffer depth (10% of injection rate and 16-flit packet size).

Similar results are obtained when measuring the impact of the variations in the average buffer occupancy. All results display average occupancy of all top ports buffers of the NoC for each scenario. Although values for the bottom ports are also available, results varied only quantitatively and were omitted. As the first column of Figure 29 shows, increasing the injection rate, the buffer depth or the packet size does not have a predictable impact in buffer occupancy. This is explained by the fact that such variations influence in other variables such as network and application latency, which end up affecting buffer occupancy estimation.



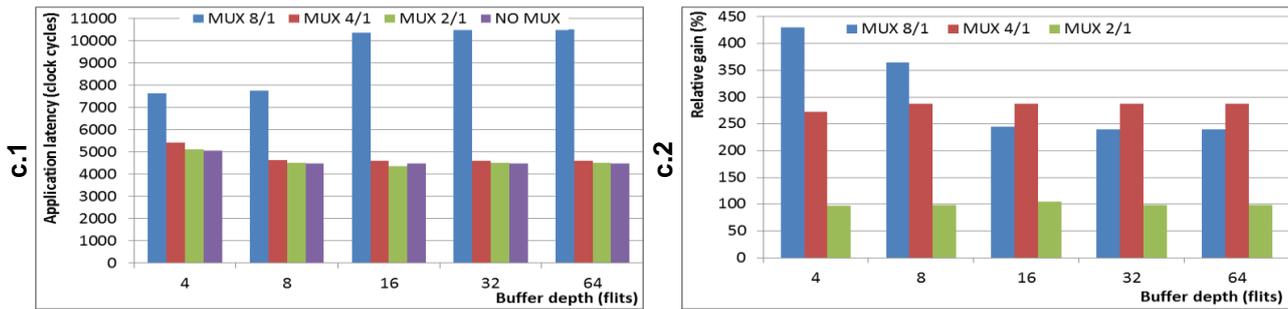


Figure 30. Total application latency (first column) and relative gains (second column) for the following variations: (a) packet size (10% of injection rate and 8-flit buffer depth); (b) injection rate (16-flit packet size and 8-flit buffer depth); (c) buffer depth (10% of injection rate and 16-flit packet size).

It is important to notice that bigger injection rates applied to different serialization schemes do not meaningfully increase buffer occupancy, suggesting that occupancy is not proportionally increased with TSV reduction. Moreover, in experiments with the MUX 8/1 scheme, the bigger the injection rate is, the bigger is the relative gain, indicating reduced occupancy figures. On the other hand, as Figure 28 and Figure 30 show, the average network latency and total application latency increase for this implementation, which leads to the misleading impression that buffer occupancy reduction is a positive result. In fact, this result only supports the statement that there is saturation point for the advantages of employing serialization schemes.

Finally, Figure 30 summarizes the impact of the simulated scenarios in the total application latency. These values were measured as the total time for executing the application for each scenario. As Figure 30(a.1), Figure 30(b.1) and Figure 30(c.1) show, for schemes MUX 2/1 and MUX 4/1, independent of the injection rate, of the buffer size and of the packet size, the total application time is always the same, or very similar to the scenario without serialization. For most cases of the MUX 8/1 scheme, total application latency increases, but the improvements on number of TSVs reduced are more substantial. In fact, as the second column of Figure 30 shows, the relative gains for all serialization levels are elevated, indicating that the advantages of serializing vertical links overcome the drawbacks that could arise in system performance.

In order to expand even more the evaluations, it is analyzed the impact of serializing TSVs on the average network latency, on the total application latency, and on the average occupancy of the buffers of top and bottom ports for different application sizes. For such simulations, an injection rate of 20% and *All-to-All* synthetic traffic were employed.

Besides, the next results presented here assume that Lasio has the following configuration: (i) *flit size: 16*; (ii) *buffer depth: 8*; (iii) *packet size: 8*. This scenario was simulated using three different application sizes: 378, 3906 and 7938 flits. From application size, it is customizable how many packets per router are required to transmit the whole application. This number is given by $(App / (Pk - 2))$, where App and Pk are the application size and packet size, respectively. In addition, four serialization schemes were evaluated: 1/1, 2/1, 4/1 and 8/1.

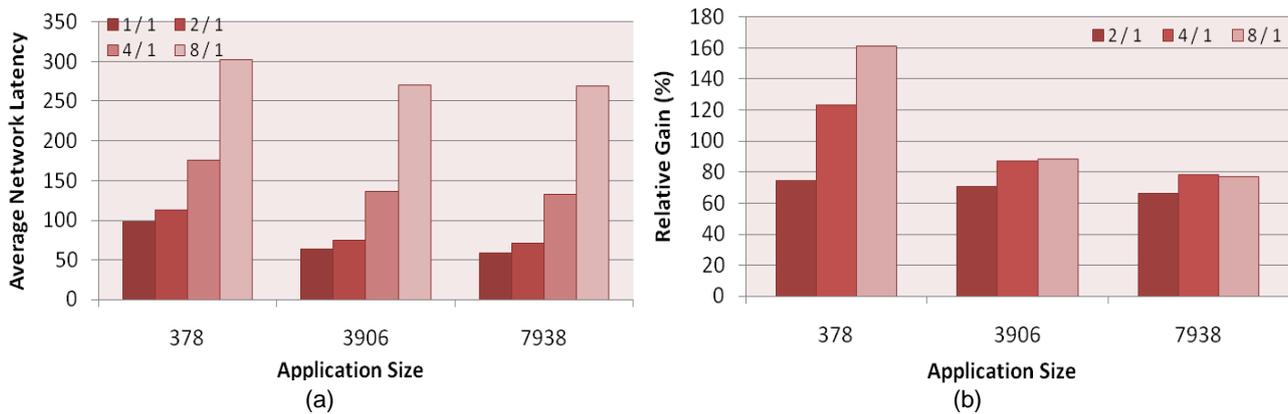


Figure 31. Average network latency, measured in clock cycles (a), and relative gain (b) for varying serialization schemes.

Figure 31 shows the results for the average network latency and the relative gains, which was measured as the expected losses with serialization scheme divided by the measured latency overhead. For instance, in Figure 31 (b) for an 8/1 serialization scheme, it is expected 8 times more latency, which is divided by the measured latency. In this way, it is possible to evaluate the tradeoffs of using serialization schemes.

As Figure 31(a) shows, for the three simulated application sizes, a 2/1 serialization scheme presented average network latency figures always close those obtained by the scheme without serialization (1/1). On the other hand, when compared to 1/1, the other schemes (i.e. 4/1 and 8/1) increase the average network latency as application size is increased. In fact, Figure 31(b) shows that the relative gains reflect these results. As the charts show, the 2/1 scheme presents rather constant gains of roughly 70%, which means that albeit network latency was increased, this increasing was not more significant than the reductions in TSV usage provided by the 2/1 scheme.

When analyzing the 4/1 and the 8/1 schemes, even though the presented relative gains are bigger than the ones presented in the 2/1 scheme, their average network latency, as Figure 31(a) shows, is bigger, which indicate a tradeoff. However, the chart

also shows that for 4/1 and 8/1 schemes, the relative gains considerably decrease as application size increases, whereas for 2/1 scheme the gains are fairly maintained. This suggests the lack of scalability for 4/1 and 8/1 schemes for varying application sizes.

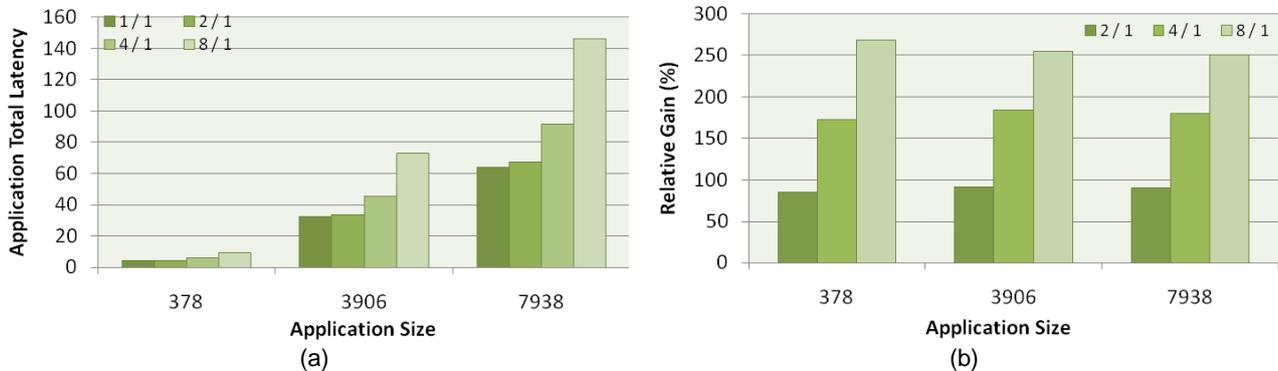


Figure 32. Application total latency, measured in clock cycles (a), and relative gain (b) for varying serialization schemes.

Figure 32 shows results for application total latency and relative gains. Figure 32(a) displays that as the application size increases; its total latency also increases. In addition, for all simulated application sizes, employing serialization implies in bigger application latencies. However, differently from the previous analysis, as Figure 32(b) shows, all simulated serialization schemes had constant relative gains for the simulated application sizes. This means that increasing in total latency is somewhat proportional to increasing in application size, no matter the serialization scheme. In this context, tradeoffs between serializing TSVs and increasing in application total latency are also observed for the simulated scenarios. Although more serialization provides more relative gains, it also provides more application latency.

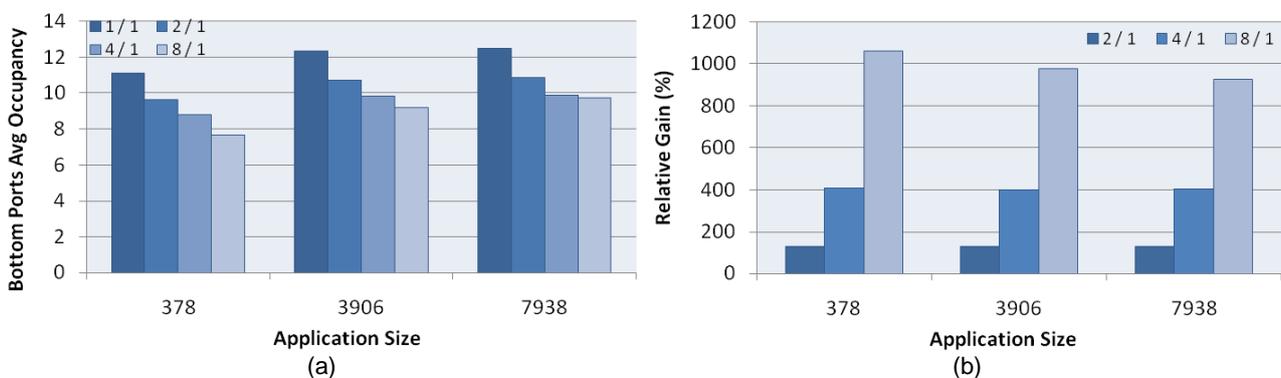


Figure 33. Bottom ports average occupancy(a) and relative gain (b) for varying serialization schemes.

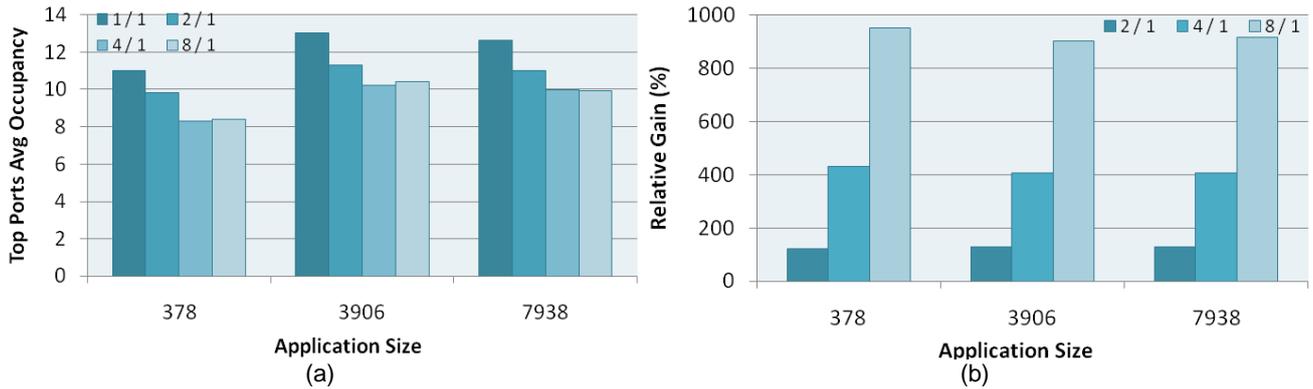


Figure 34. Top ports average occupancy(a) and relative gain (b) for varying serialization schemes.

Figure 33 and Figure 34 present results for bottom and top inputs buffers occupancy, respectively. The results presented in Figure 33(a) and Figure 34(a) were obtained by computing the average occupancy (in %) of all bottom and top inputs buffers during simulation until the last packet of the application was received.

As the charts show, bigger applications cause slightly bigger occupancy figures, because there are more packets transmitted during execution. In addition, the reduction of occupancy observed when employing serialization schemes is justified by the increase in application total latency, showed in Figure 32, because the execution time is increased. That is also the reason for the expressive relative gains observed in Figure 33(b) and Figure 34(b).

7.3.2 AREA CONSUMPTION AND POWER DISSIPATION

In order to evaluate the area consumed by the proposed serialization scheme, Lasio was synthesized targeting the 65 nm STMicroelectronics CMOS technology. Accordingly, four versions of the central router (contains all seven ports) were synthesized: 2/1, 4/1 and 8/1 serialization and without serialization. The synthesis was done using Cadence RTL Compiler and employed a general-purpose standard cell library provided by the foundry. Additionally, the implementations were designed for operating at 1 GHz. After technology mapping, a statistical power analysis was performed for comparing the power overhead of the serialization schemes on the central router. Table 5 summarizes the obtained results for comparing the synthesized routers, assuming flit size of 16 bits and buffer depth of 8 flits.

Table 5. Standard cells area and power results for lasio synthesis with four serialization schemes. Static power stands for the measured leakage power of standard cells and dynamic power assumes 50% of switching activity.

Serialization scheme	Area (mm²)	Static Power (mW)	Dynamic Power (mW)
non	0.0784	0.988	42.43
2 / 1	0.0792	0.999	55.54
4 / 1	0.0793	0.997	55.55
8 / 1	0.0796	1.001	55.69

Observing Table 5, there is very little overhead in standard cell area when employing the serialization schemes, fewer than 2%. Moreover, the difference between each serialization scheme is even lower and under 1%, which indicates the scalability of the method.

Static power results also point to slight overheads when adopting a serialization scheme, under 2% in the worst case. Additionally, similarly to the area results, the overhead between the implementations that adopted serialization is negligible, displaying good scalability. Dynamic power, on the other hand, displayed an overhead of over 30% in the worst case. However, the reductions in TSV usage may justify this overhead. Thus, a tradeoff is presented, where the technique can be employed for reducing TSVs, if it meets power restrictions. In addition, the obtained results suggest that the difference in dynamic power dissipation between the implementations that employing the serialization scheme is also negligible.

8 CONCLUSIONS AND FUTURE WORKS

NoC have been successfully employed as a solution to deal with communication in complex MPSoC. NoC-based architectures are characterized by various tradeoffs related to structural characteristics, performance specifications, and application demands. In new technologies, the relative values of wire delays and power dissipation are increasing as the number of cores in 2D chips increase. The recent 3D IC technology allows better performance enhancements with less scaling concerns. 3D integration applied to NoC architectures permits greater device integration, smaller chip paths, higher communication bandwidth, and shorter interconnection links, which directly influences the communication performance representing fundamental role in the design of high performance MPSoC. In the present work, evaluations were performed in order to explore 3D NoCs configurations that might contribute to an improved communication performance.

During this work development, an extensive work review was performed in publications related to 3D NoCs, focusing mainly on 3D NoCs architectural explorations and on efforts on design techniques of using TSV as vertical communication links between stacked layers, its viability and its drawbacks. Based on the studies analyzed, varied 3D NoCs configurations were implemented and evaluated modifying network and architectural parameters to point out and clarify general and punctual benefits and challenges. Moreover, a TSV serialization scheme was explored in order to reduce the interlayer links, alleviating the bottlenecks related to interconnection problems.

To achieve such accomplishments, this Dissertation proposed a 3D NoC called Lasio, whose implementation was based on Hermes 2D NoC [MOR04]. This work addresses a wide 3D NoC architectural exploration. The contributions comprise three main topics:

- Topological impact on latency and throughput comparing 3D NoC and 2D NoC;
- Buffer depth and traffic influence on 3D NoCs performance directing to a TSV serialization scheme viability;
- TSV serialization scheme analyzing area occupation, power dissipation and overall performance.

The topological impact on latency and throughput comparing 3D NoC and 2D NoC is based on an evaluation of network and application latencies and throughputs on Hermes and Lasio. Lasio implements all architectural features of Hermes, which enable a rational comparison between NoC topologies. Both topologies were compared according to several buffer depth, packet sizes and two traffic scenarios. For the selected set of experiments, 3D NoC implementation minimized the latencies of the packets when compared to 2D NoC implementation. For instance, the average NoC latency minimization was 25% and the application latency minimization was 30%, on average. Another assessment was on the 3D NoC Lasio and the influence of several buffer depths and packet sizes on network and application latencies. Results demonstrate that when applying an appropriate buffer depth both latencies are reduced as well as the throughput is increased.

Next, in the buffer depth and traffic influence on 3D NoCs performance, it was analyzed the occupancy of a study case of Lasio. Results show that buffers of top and bottom ports are typically underused, which makes opportune to explore TSV serialization schemes. These schemes might also be implemented at different layer levels depending on the employed routing algorithm and traffic scenario. TSV serialization enables higher design space exploration, in order to reduce power dissipation and area consumption of 3D NoCs. In this sense, results indicate that the technique can help coping with technological issues, especially the limited number of TSVs in a 3D system.

Finally, this work proposed and evaluated a vertical links (i.e. TSVs) serialization scheme for Lasio 3D NoC and the impact of such serialization in terms of area consumption, power dissipation and overall performance. The obtained results suggest that, albeit there are some losses in network latency, the proposed serialization scheme still presents significant relative gains, as its improvements in terms of TSV reduction are more substantial than the reported losses. In addition, lower losses were observed for 4/1 and 2/1 serialization schemes, whereas for 8/1 scheme these losses were bigger, indicating a saturation point for the benefits of employing serialization. Regarding area consumption and static power dissipation, there is no significant overheads when adopting the serialization technique, under 2%, in worst case. However, there is a significant increasing in dynamic power dissipation of over 30%, in worst case. In this way, a tradeoff between number of TSVs and dynamic power dissipation is presented for the serialization

technique. Therefore, the reported results suggest that the proposed scheme is well suited to cope with 3D IC era challenges.

8.1 Future Works

This work is an initial effort to propose, implement and evaluate a 3D NoC with parameterizable configurations. Improvements on the Lasio 3D NoC may be addresses as future works comprised in four main topics:

- Evaluate traffic scenarios variations (Section 6.1.3);
- Assess different routing algorithms and their impact on TSV serialization;
- Evaluations concerning TSVs quantity and placement in a 3D integration;
- Employment of real applications to evaluate the proposed schemes for 3D NoCs.

The first future work that might be executed is to evaluate five variations of traffic scenario described in Section 6.1.3 – *All-to-All-Next*, *All-to-All-Complement*, *All-to-Bottom*, *All-to-Top* and *Random*. *All-to-All-Next* and *All-to-All-Complement* analysis can help diminishing the output data congestion as the traffic increases. Besides, *All-to-All-Complement* performs a different traffic distribution enabling to explore scenarios in terms of high data transmission in vertical links. The main idea of *All-to-Bottom* and *All-to-Top* scenarios is to overload vertical communication channels. Possibly, either *All-to-Bottom* or *All-to-Top* combined with the TSV serialization scheme will improve overall communication performance, since vertical communication channels are typically underused [GHI13a]. Finally, *Random* scenario might be useful to determine possible congestion areas in a 3D NoC communication pattern due to its traffic unpredictability, indicating, for instance, where are the best places to position a TSV, how many TSVs are necessary, and the level of its serialization.

Assess different routing algorithms and their impact on TSV serialization includes the bottom-first routing algorithm analysis. This algorithm utilizes primarily vertical links during the communication, which means, that it will demonstrate the TSV communication capacity. Moreover, more than one routing algorithm might be implemented in each router. This is an interesting approach – routers are able to make a choice among varied routing algorithms in runtime according to the packet path, embracing on algorithm over another

one. Such implementation will expend more router area; however, overall NoC performance improvements might also be expected.

With TSV serialization consolidated, evaluations concerning TSVs quantity and placement become a motivating topic, since TSVs consume a relative large amount of chip area and are error-prone during manufacturing resulting in yield reduction for large TSV counts. Besides, data transmission in 3D NoC exhibits a temporal characteristic that TSVs in different nodes are rarely busy at the same time [LIU11], which also allows less TSVs in a 3D IC. The idea of reducing the number of TSVs while maintaining interconnection performance might be addressed by operating TSVs at much higher frequencies than intra-layer interconnections and serializing several inter-layer links. Rahmani et al. [RAH11] approached this issue through a TSV array operating in a bidirectional manner. Moreover, frequency upscaling is applied to compensate the throughput loss compared to unidirectional links. Notwithstanding, some complexity in the switching logic may be expected due to the reduction of TSV count.

As a finally future work, it is proposed the employment of real applications as case studies to verify tradeoffs on the 3D NoC architectural exploration and on the serialization schemes. However, such evaluation of real applications encompasses the research of two other related activities: (i) the partitioning of tasks in groups and (ii) the mapping of task-groups in PEs targeting 3D NoC-based MPSoCs.

REFERENCES

- [BAN01] K. Banerjee et al. **3-d ics: A novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration**. *Proceedings of IEEE*, v. 89, Issue 5, May 2001, pp. 602-633.
- [BEN02] L. Benini, G. De Micheli. **Networks on Chips: A New SoC Paradigm**. *IEEE Computer*, v. 35, Issue 1, Jan. 2002, pp. 70-78.
- [BER07] K. Bernstein et al. **Interconnects in the Third Dimension: Design Challenges for 3D ICs**. *Design Automation Conference (DAC)*, 2007, pp. 562-567.
- [BLA04] B. Black et al. **3D Processing Technology and its Impact on IA32 Microprocessors**. *IEEE International Conference on Computer Design (ICCD)*, 2004, pp. 316-318.
- [BRU09] J. Bruch et al. **BrownPepper: A SystemC-based simulator for performance evaluation of Networks-on-Chip**. *Very Large Scale Integration (VLSI-SoC)*, 2009, pp. 223-226.
- [BUT11] M. Buttrick et al. **Mitigating Partitioning, Routing, and Yield Concerns in 3D ICs by Multiplexing TSVs**. *IEEE Computer Society Symposium on VLSI (ISVLSI)*, 2011, pp. 194-199.
- [CHA04] J. Chan et al., **“NoCGEN: A Template Based Reuse Methodology for Networks on Chip Architecture”**, *VLSI Design (VLSID)*, 2004, pp. 717-720.
- [CHA08] J. Charbonnier et al. **Wafer level packaging technology development for CMOS image sensors using Through Silicon Vias**. *Electronics System-Integration Technology Conference (ESTC)*, 2008, pp. 141-148.
- [CUL98] D. Culler, J. Singh. **Parallel Computer Architecture: a Hardware Software Approach**. *Morgan Kaufmann*, Los Altos, USA, ed. 1, 1998, 1100 p.
- [DAL01] W. Dally, B. Towles. **Route Packets, Not Wires: On Chip Interconnection Networks**. *Design Automation Conference (DAC)*, 2001, pp. 684-689.
- [DAL86] W. Dally, C. Seitz. **The Torus Routing Chip**. *Journal of Distributed Computing*, v. 1, Issue 4, 1986, pp. 187-196.
- [DAL04] W. Dally, B. Towles. **Principles and Practices on Interconnection**

Networks, Morgan Kaufmann, 1st edition, Austin, Texas, 2004, 550 p.

- [DON09] X. Dong, Y. Xie. **System-level cost analysis and design exploration for three-dimensional integrated circuits (3D ICs)**, *Asia and South Pacific Design Automation Conference (ASPDAC)*, 2009, pp. 234-241.
- [EMB13] Embedded Systems Development. Available at www.embedded.com, 2013.
- [FEE07] B. Feero, P. Pande. **“Performance evaluation for three-dimensional networks-on-chip.** *IEEE Computer Society Symposium on VLSI (ISVLSI)*, 2007, pp. 305-310.
- [FEE09] B. Feero, P. Pande. **Networks on Chip in a Three Dimensional Environment: A Performance Evaluation.** *IEEE Transactions on Computers*, vol. 58, Issue 1, Jan. 2009, pp. 32-45.
- [GAR08] P. Garrou et al. **Handbook of 3D Integration: Technology and Applications of 3D Integrated Circuits**, *Wiley Online Library*, Online Edition, Weinheim, Germany, 2008, 773 p.
- [GHI12a] Y. Ghidini et al. **Topological Impact on Latency and Throughput: 2D versus 3D NoC Comparison.** *Symposium on Integrated Circuits and Systems Design (SBCCI)*, 2012, pp. 1-6.
- [GHI12b] Y. Ghidini et al. **Buffer Depth and Traffic Influence on 3D NoCs Performance.** *IEEE International Symposium on Rapid System Prototyping (RSP)*, 2012, pp. 9-15.
- [GHI13a] Y. Ghidini et al. **TSV Multiplexing: A 3D NoC Occupancy Analysis.** *Design, Automation & Test in Europe (DATE)*, 2013.
- [GHI13b] Y. Ghidini et al. **Lasio 3D NoC Vertical Links Serialization: Evaluation of Latency and Buffer Occupancy.** *Symposium on Integrated Circuits and Systems Design (SBCCI)*, 2013, pp. 1-6.
- [GRA11] M. Grange et al. **Modeling the Computational Efficiency of 2-D and 3-D Silicon Processors for Early-Chip Planning.** *International Conference on Computer-Aided Design (ICCAD)*, 2011, pp. 310-317.
- [HEN07] D. Henry et al. **Via First Technology Development Based on High Aspect Ratio Trenches Filled with Doped Polysilicon.** *Electronic Components and Technology Conference (ECTC)*, 2007, pp. 830-835.

- [HOR01] M. Horowitz et al. **The Future of Wires**. *Proceedings of IEEE*, vol. 89, n. 4, Apr. 2001, pp. 490-504.
- [ITR07] International Technology Roadmap for Semiconductors: Interconnect. Available at www.itrs.net/, 2007.
- [ITR12] International Technology Roadmap for Semiconductors: Interconnect. Available at www.itrs.net/, 2012.
- [JAN03] A. Jantsch, H. Tenhunen. **Network on Chip**. *Kluwer Academic Publishers*, ed. 1, 2003, 301 p.
- [KAW83] S. Kawamura, et al. **Three-dimensional CMOS IC's fabricated by using beam recrystallization**. *IEEE Electron Device Letters*, vol. 4, Issue 10, Oct. 1983, pp. 366-368.
- [LI06] F. Li et al **Design and management of 3D chip multiprocessors using network-in-memory**. *International Symposium on Computer Architecture (ISCA)*, 2006, pp. 130-141.
- [LIU11] C. Liu et al. **Vertical Interconnects Squeezing in Symmetric 3D Mesh Network-on-Chip**. *Asia and South Pacific Design Automation Conference (ASPDAC)*, 2011, pp. 357-362.
- [LOI08] I. Loi et al. **A low-overhead fault tolerance scheme for TSV-based 3D network on chip links**. *International Conference on Computer-Aided Design (ICCAD)*, 2008, pp. 598-602.
- [LOI11] I. Loi et al. **Characterization and Implementation of Fault-Tolerant Vertical Links for 3-D Networks-on-Chip**. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, v. 30, Issue 1, Jan. 2011, pp. 124-134.
- [MAR05] C. Marcon, **Modelos para o Mapeamento de Aplicações em Infraestruturas de Comunicação Intrachip**. *Universidade Federal do Rio Grande do Sul*, Tese de Doutorado em Ciência da Computação, Porto Alegre, 2005, 192 p.
- [MAR12] E. Marinissen. **Challenges and Emerging Solutions in Testing TSV-Based 2 1/2D- and 3D-Stacked ICs**. *Design, Automation & Test in Europe (DATE)*. 2012, pp. 1277-1282.

- [MIL13] F. Miller et al. **Virtualized and Fault-Tolerant Inter-Layer-Links for 3D-ICs.** *Microprocessors and Microsystems*, v. 7, Issue 8, Nov. 2013, pp. 823-835.
- [MOH98] P. Mohapatra. **Wormhole Routing Techniques for Directly Connected Multicomputer Systems.** *ACM Computing Surveys*, v. 30, Issue 3, Sep. 1998, pp. 374-410.
- [MOR04] F. Moraes et al. **HERMES: An Infrastructure for Low Area Overhead Packet-switching Networks on Chip.** *Integration, the VLSI Journal*, v. 38, Issue 1, Oct. 2004, pp. 69-93.
- [MOR10] E. Moreno. **Mapeamento e Adaptação de Rotas de Comunicação em Redes em Chip.** *Pontifícia Universidade Católica do Rio Grande do Sul*, Tese de Doutorado em Ciência da Computação, Porto Alegre, 2010, 177 p.
- [MUT00] J. Muttersbach et al. **Practical design of globally-asynchronous locally-synchronous systems.** *International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC)*, 2000, pp. 52-59.
- [NAK84] M. Nakano. **3-D SOI/CMOS.** *International Electron Device Meeting (IEDM)*, 1984, pp. 792-795.
- [OGR07] U. Ogras et al. **Challenges and promising results in NoC prototyping using FPGAs.** *IEEE Micro*, vol. 27, Issue 5, Sep. 2007, pp. 86-95.
- [OST05] L. Ost et al. **MAIA - A Framework for Networks on Chip Generation and Verification.** *Asia and South Pacific Design Automation Conference (ASPDAC)*, 2005, pp. 49-52.
- [PAP11] A. Papanikolaou et al. **Three Dimensional System Integration: IC Stacking Process and Design.** *Springer*, 2011 edition, 2011, 243 p.
- [PAR08] D. Park et al. **MIRA: A Multi-Layered On-Chip Interconnect Router Architecture.** *IEEE International Symposium on Computer Architecture (ISCA)*, 2008, pp. 251-261.
- [PAR09] G. Pares et al. **Mid-process through silicon vias technology using tungsten metallization: Process optimization and electrical results.** *Electronics Packaging Technology Conference (EPTC)*, 2009, pp. 772-777.
- [PAS09] S. Pasricha. **Exploring Serial Vertical Interconnects for 3D ICs.** *Design Automation Conference (DAC)*, 2009, pp. 581-586.

- [PAT06] R. Patti. **Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs**. *Proceeding of the IEEE*, v. 94, Issue 6, 2006, pp. 1214-1224.
- [PAV06] V. Pavlidis, E. Friedman. **Three-Dimensional (3-D) Topologies for Networks-on-Chips**. *IEEE International System-on-Chip Conference (SOCC)*, 2006, pp. 285-288.
- [PAV07] V. Pavlidis, E. Friedman. **3-D Topologies for Networks-on-Chip**. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, v. 15, Issue 10, Oct. 2007, pp. 1081-1090.
- [PAV08] V. Pavlidis, E. Friedman. **Three-dimensional Integrated Circuit Design**, *Morgan Kaufmann*, 2008 edition, 2008, 284 p.
- [RAH10] A. Rahmani et al. **Research and Practices on 3D Networks-on-Chip Architectures**, *Norchip*, 2010, pp. 1-6.
- [RAH11] A. Rahmani et al. **Lastz: An ultra-optimized 3d networks-on-chip architecture**, *Euromicro Conference on Digital System Design (DSD)*, 2011, pp. 173-180.
- [RAH13] A. Rahmani et al. **Developing a power-efficient and low-cost 3D NoC using smart GALS-based vertical channels**. *Journal of Computer and System Sciences*, v. 79, n. 4, Jun. 2013, pp. 440-456.
- [RAM09] R. Ramanujam, B. Lin. **A Layer-Multiplexed 3D On-Chip Network Architecture**, *IEEE Embedded Systems Letters*, v. 1, Issue 2, Oct. 2009, pp. 50-55.
- [SAV00] S. Savastiouk, **Moore's Law – the z-dimension**, *Solid State Technology*, vol. 43, Issue 1, Dec 2000, 84 p.
- [SPI04] S. Spiesshoefer et al. **Z-axis interconnects using fine pitch, nanoscale through-silicon vias: Process development**. *Electronic Components and Technology Conference (ECTC)*, 2004, pp. 466-471.
- [SUN05] V. Suntharalingam et al. **Megapixel CMOS image sensor fabricated in three-dimensional integrated circuit technology**. *IEEE International Solid-State Circuits Conference (ISSCC)*, 2005, pp. 356-357.
- [SUN10] F. Sun et al. **Design and Feasibility of Multi-Gb/s Quasi-Serial Vertical**

- Interconnects based on TSVs for 3D ICs.** *VLSI System on Chip Conference (VLSI-SoC)*, 2010, pp. 149-154.
- [TOP06] A. Topol, et al. **Three-dimensional integrated circuits.** *IBM Journal of Research and Development*, v. 50, n. 4, 2006, pp. 491-506.
- [XU10] T. Xu et al. **A Study of Through Silicon Via Impact to 3D Network-on-Chip Design.** *International Conference on Electronics and Information Engineering (ICEIE)*, 2010, pp. 333-337.
- [VIV11] P. Vivet et al. **3D NoC using through silicon Via: An asynchronous implementation.** *VLSI and System-on-Chip (VLSI-SoC)*, 2011, pp. 232-237.
- [WOL08] M. Wolf et al. **Technologies for 3D wafer level heterogeneous integration.** *Symposium on Design, Test, Integration and Packaging of MEMS/MOEMS (DTIP)*, 2008, pp. 123-126.
- [YIN11] A. Yin et al. **Change Function of 2D/3D Network-on-Chip.** *Computer and Information Technology (CIT)*, 2011, pp. 181-188.
- [YON11] J. Yong et al. **3D Network-on-Chip System Communication Using Minimum Number of TSVs.** *International Conference on ICT Convergence (ICTC)*, 2011, pp. 517-522.
- [ZIA11] A. Zia et al. **3D NoC for many-core processors.** *Microelectronics Journal*, v. 42, n. 12, 2011, pp. 1380-1390.
- [ZIP04] P. Zipf et al. **A Switch Architecture and Signal Synchronization for GALS System-on-Chips.** *Symposium on Integrated Circuits and System Design (SBCCI)*, 2004, pp. 210-215.

APPENDIX A: LASIO 3D NOC REGISTER TRANSFER LEVEL (RTL)

Lasio 3D Network on Chip, its mechanisms and resources are all described in VHDL files (detailed below). Its configurable parameters, such as the definition of the network dimensions, flits width, buffers depth, routing algorithms, packet sizes, injection rates, application sizes, TSV serialization level, and traffic scenario are selected from Electra and implemented in these VHDL output files:

- *Lasio_package*: contains Lasio network specific library. In this library are inserted the parameterizable values selected by the user from Electra. This values definition (e.g., flit size, TSV size, and number of routers) is utilized by all others VHDL files.
- *Lasio_buffer*: implements a queue (buffer) for provisional storage in each routers input port, reducing the routers affected by the flits blockade, which is performed by the switching algorithm. Depth queues is parameterizable from *Lasio_package* file and the queues are implemented as circular FIFOs.
- *Lasio_crossbar*: performs the switching between routers ports. Utilizes and manipulates signals to indicate which ports are interconnected verifying whether there is or not data to be transmitted and the ports availability. Moreover, all ports (Local, North, South, East, West, Top and Bottom) connections are described in this file.
- *Lasio_switchcontrol*: this file describes the routing algorithm and the arbiter task that analyzes the content of the packet, verify its destination and choose the direction and the path to be traversed.
- *Router*: contains the instantiation of the files described above. Routers perform the messages transfer between cores and they have control logic implemented responsible for routing and seven bidirectional ports.
- *NOC*: Instantiate all the routers and establishes the connection between them. The number of routers created depends on the NoC dimension, which is parameterizable. The Local port establishes the communication between the router and its core. The other ports connect each router to its neighboring routers.
- *TopNoC*: NoC testbench. It instantiates the entire NoC and its parameters selected from Electra. Besides, it generates clock signals for the routers and

instantiate the NoC. *TopNoC* file is also responsible to produce report output files during the simulation (Appendix B);

APPENDIX B: REPORT OUTPUT FILES

During the simulation, Lasio generates report output files that contain all packets information (path traveled, time spent in each router, number of hops) to evaluate parameters, such as latency and throughput. These output files also contain buffers information, like input and output time of a packet in order to assess the buffer occupancy. Such data are all measured by clock cycle and packet-by-packet. The following files are the ones generated and analyzed in this Dissertation:

- *File_io.out*: this file holds all packets information regarding its path traveled and the time each packet spent from its source to its target. This report is used to appraise parameters such as, latency and throughput. Data are extracted in the following format (Figure 35):

Target_address	Packet_size	Source_address	App_input_time	NoC_input_time	NoC_output_time
----------------	-------------	----------------	----------------	----------------	-----------------

Figure 35. *File_io.out* file extracted data.

- *Target_address* – contains the destination router address in “XYZ” coordinates;
 - *Packet_size* – comprises the size of the packet (flits number);
 - *Source_address* – contains the source router address in “XYZ” coordinates;
 - *App_input_time* – It is the planned injection time of a packet into the NoC (see Figure 20);
 - *NoC_input_time* – It is the accomplished injection time of a packet - considers the exact insertion timing of a packet into the NoC (see Figure 20);
 - *NoC_output_time* – It is the packet accomplished reception time by the router. It shows the real delivery time of packets at their destinations (see Figure 20).
- *Pkt_flow.out*: This file shows, in each clock cycle, how many flits are inside the network searching for their destinations and how many flits have already reached their final target. This report is used to measure buffers occupancy;

- *Buffer_<xyyzz>.out*: The number of the *Buffer_<xyyzz>.out* files produced is the same of the number of routers. In each of these files, it is shown all the seven buffers corresponding to the seven input ports and what is the buffer depth occupied by the flits. This report is used to assess buffers and links occupancy and to identify possible bottlenecks/congestions inside the network.