UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

DÉBORA DA SILVA MOTTA MATOS

Exploring Hierarchy, Adaptability and 3D in NoCs for the Next Generation of MPSoCs

Tese apresentada como requisito parcial para a obtenção do grau de Doutora em Ciência da Computação.

Advisor
Prof. Dr. Altamiro Amadeu Susin

Co-advisor
Prof. Dr. Luigi Carro

Porto Alegre
2014

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Matos, Débora da Silva Motta

Exploring Hierarchy, Adaptability and 3D in NoCs for the Next Generation of MPSoCs [manuscrito] / Débora da Silva Motta Matos. – 2014.

200 f.:il.

Orientador: Altamiro Amadeu Susin; Co-orientador: Luigi Carro.

Tese (doutorado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2014.

1. Networks-on-Chip. 2. Hierarchical topology 3. Adaptability 4. Three dimensional NoCs 5. Hardware I. Susin, Altamiro Amadeu. II. Carro, Luigi. III. Exploring Hierarchy, Adaptability and 3D in NoCs for the Next Generation of MPSoCs.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Carlos Alexandre Netto
Vice-Reitor: Prof. Rui Vicente Oppermann
Pró-Reitor de Pós-Graduação: Prof. Vladimir Pinheiro do Nascimento
Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb
Coordenador do PPGC: Prof. Luigi Carro
Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Inicio meus agradecimentos à pessoa mais especial em minha vida que me apoia e me incentiva todos os dias, que compreendeu os momentos em que precisamos ficar meses separados para que eu pudesse realizar colaborações com outras instituições de pesquisa que foram tão importantes para minha formação: meu amado esposo Cristiano. A ti um agradecimento especial, pela tua eterna compreensão em todos os momentos, pelo teu carinho e por aceitar e entender a minha "falta" que se tornaram uma constância na etapa final deste trabalho. Obrigada por fazer parte da minha vida!

Agradeço Àquele que está presente em todos os momentos da minha vida, guiando minhas atitudes e me dando forças: Deus.

Ao meu orientador, Prof. Altamiro Susin, que muito admiro pelo seu profissionalismo e amizade. Por sempre estar disponível quando precisei de uma orientação, de um apoio ou apenas de uma palavra amiga. Obrigada pela sua contribuição no desenvolvimento desse trabalho e por seu apoio constante ao longo de todos esses anos.

Ao meu co-orientandor, que também admiro bastante, Luigi Carro, que muito me incentivou durante esse período, me instigando a fazer sempre mais e melhor, a encontrar ideias inovadoras, a obter um diferencial e a pensar como uma cientista. Obrigada pela revisão de todos os meus trabalhos e artigos, pelas várias contribuições e incontáveis dicas. Obrigada pela compreensão e apoio em todos os momentos.

À minha mãe, Dulce, que é uma das responsáveis pela pessoa que sou e pelos valores e princípios que trago comigo, que sempre esteve disposta a me ajudar e compreender os meus momentos de ausência. Agradeço também a minha sogra, pela compreensão e por sempre se mostrar disposta a me ajudar com as demais tarefas para que me sobrasse mais tempo no desenvolvimento das minhas atividades acadêmicas.

Um forte agradecimento a minha amiga e colega Caroline Concatto, com quem realizei diversos trabalhos e publicações e que sempre esteve disposta a trocar uma ideia sobre a nossa pesquisa ou somente jogar conversa fora. Agradeço também a Anelise Kologeski pelas várias discussões de trabalho, pelo período em que dividimos apartamento em Milão e todos os lugares que descobrimos naquela época.

Agradeço também a todos os colegas do Lab215 e amigos que fiz no PPCG e PGMICRO, que ao longo desses 6 anos (mestrado e doutorado) foram muitos os que

por lá passaram, e sendo assim, abstenho-me de listar todos os nomes, pois certamente me esqueceria de alguém que, de um certo modo, participou das incontáveis conversas que tivemos, seja sobre a pesquisa, ou sobre assuntos aleatórios.

Alguns agradecimentos que foram realmente MUITO importantes para o trabalho específico da tese. Começo com o meu amigo Márcio Kreutz, que sempre esteve disponível para começarmos novas pesquisas e desenvolvermos novas ideias. Márcio, tu certamente sabes que essa tese só existe por causa da tua enorme ajuda em tantos resultados e análises que fizemos durante esse período. Obrigada pela colaboração em todos os artigos que escrevemos durante esse período. Que venham muito mais! Muito obrigada por tudo, por todas as conversas diárias. Admiro-te muito pela tua tranquilidade e otimismo, preciso aprender mais sobre isso contigo.

Outro agradecimento importante é ao Cezar Reinbrecht que também colaborou em muitas etapas desse trabalho, um trabalho que teve uma evolução bem interessante, sendo discutido e realizado passo-a-passo. Obrigada por confiar na minha pesquisa e nas minhas ideias. Agradeço ainda aos bolsistas de IC que tive durante esse período e que também colaboraram com esse trabalho: Rafael Leite, Paulo Cesar, Jonathan Martinelli e Tiago Motta. Esse último, além de meu irmão, contribuiu muito na obtenção dos últimos resultados. Agradeço-te pela tua compreensão nos últimos dois meses antes da defesa da tese.

Agradeço aos meus supervisores durante os estágios em Milão e Grenoble, Gianluca Palermos e Fabien Clermidy pelo grande aprendizado que obtive durante estas colaborações. Agradeço ainda aos demais professores do PPGC que contribuíram na minha formação: Fernanda Kastensmidt, Leila Ribeiro, Erika Cota, Sergio Bampi, dentre outros, e agradeço aos professores Cesar Zeferino, Monica Pereira e Flávio Wagner pelas importantes observações feitas sobre essa tese. Ao CNPq, agradeço por financiar meu período de estudos durante o doutorado.

Aos demais amigos e familiares que estão sempre torcendo por mim para que eu alcance meus objetivos e a todas as pessoas brilhantes que conheci desde que ingressei nesta Universidade: um muito obrigada!

Explorando Hierarquia, Adaptabilidade e 3D em NoCs para a Próxima Geração de MPSoCs

RESUMO

A demanda por sistemas com elevado desempenho tem trazido a necessidade de aumentar o número de elementos de processamento, surgindo os chamados Sistemas em Chip Multiprocessados (MPSoCs). Além disso, com a possibilidade de redução da escala tecnológica na era submicrônica, permitindo a integração de vários dispositivos, os chips têm se tornado ainda mais complexos. No entanto, com o aumento no número de elementos de processamento, as interconexões são vistas com o principal gargalo dos sistemas-em-chip. Com isso, uma preocupação na forma como tais elementos se comunicam e estão interconectados tem sido levantada, uma vez que tais características são cruciais nos aspectos de desempenho, energia e potência, principalmente em sistemas embarcados. Essa necessidade permitiu o advento das redes-em-chip (Networks-on-Chip – NoCs) e inúmeros estudos já foram realizados para tais dispositivos. No entanto, devido ao aceleramento tecnológico atual, que traz a necessidade por sistemas ainda mais complexos, que consumam baixa energia e que permitam que as aplicações sejam constantemente atualizadas sem perder as características de desempenho, as arquiteturas de interconexão tradicionais não serão suficientes para satisfazer tais desafios. Outras alternativas de interconexão para MPSoCs precisam ser investigadas e nesse trabalho novas arquiteturas para NoCs com tais requisitos são apresentadas. As soluções propostas exploram hierarquia, adaptabilidade e interconexões em três dimensões. Esse trabalho aborda a necessidade do uso de diferentes estratégias em NoCs a fim de atingir os requisitos de desempenho e baixo consumo de potência dos atuais e futuros MPSoCs. Dessa forma, serão verificadas as diversas arquiteturas de interconexões para sistemas heterogêneos, sua escalabilidade, suas principais características e as vantagens das propostas apresentadas sobre as demais soluções.

Palavras-Chave: Rede-em-Chip, topologia hierárquica, crossbar, adaptabilidade, chaveamento, projetos 3D, interconexão 3D.

ABSTRACT

The demand for systems with high performance has brought the need to increase the number of cores, emerging the called Multi-Processors System-on-Chip (MPSoCs). Also, with the shrinking feature size in deep-submicron era, allowing the integration of several devices, chips have become even more complex. However, with the increase in these elements, interconnections are seen as the main bottleneck in many core systems-on-chip. With this, a concern about how these devices communicate and are interconnected has been raised, since these features are crucial for the performance, energy and power consumption aspects, mainly in embedded systems. This need allows the advent of the Networks-on-Chip (NoCs) and countless studies had already been done to analyze such interconnection devices. However, due to the current technological accelerating that brings the need for even more complex systems, consuming lower energy and providing constant application updates without losing performance features, traditional interconnect architectures will not be sufficient to satisfy such challenges. Other interconnecting alternatives for MPSoCs need to be investigated and in this work, novel architectures for NoCs meeting such requirements are presented. The proposed solutions explore hierarchy, adaptability and three dimensional interconnections. This work approaches the requirements in the use of different strategies for NoCs in order to reach the performance requisites and low power consumption of the current and future MPSoCs. Hence, in this approach, several interconnection architectures for heterogeneous systems, their scalability and the main features and advantages of the proposed strategies in comparison with others will be verified.

Keywords: Network-on-chip, hierarchical topology, crossbar, adaptability, switching, 3D designs, 3D interconnects.

LIST OF FIGURES

Figure 1.1 - Trend f number of PEs for SoC Consumer Portable Design Complexity. . 26
Figure 1.2 - (a) Example of Clock Skew and (b) Jitter. 27
Figure 1.3 – (a) VOPD and (b) MPEG benchmarks with the communication core graph. 28
Figure 1.4 - Example of cores in an MPSoC. 29
Figure 2.1 - Example of point-to-point interconnection. 36
Figure 2.2 - (a) Example of system interconnected by Intel QuickPath and (b) its architecture. 38
Figure 2.3 - Example of a system interconnected by a shared bus. 40
Figure 2.4 - Hierarchical AMBA bus architecture. 41
Figure 2.5 - IBM Cell ring bus communication architecture. 41
Figure 2.6 – SH Mobile G1 hierarchical bus communication architecture. 42
Figure 2.7 - Example of a full bus crossbar. 43
Figure 2.8 - Crossbar architecture: wires, multiplexers and arbiters. 43
Figure 2.9 - Example of maximum wire length according to the number of PEs interconnected by a crossbar 45
Figure 2.10 - Maximum latency according to the crossbar size considering cores with an area of 1mm². 45
Figure 2.11 - Maximum latency according to the crossbar size considering cores with an area of 0.488mm². 45
Figure 2.12 - Trend of total interconnection length on a chip. 46
Figure 2.13 - Wire delay per gate delay according to the technology scale. 48
Source: elaborated by the author. 48
Figure 2.14 - Maximum wire length at every frequency. 48
Figure 2.15 - Wire dimensions for an integrated circuit. 49
Figure 2.16 - Power consumption at different link lengths and operating frequencies. . 49
Figure 2.17 - Reasons that point out to the network-on-chip. 50
Figure 2.18 - Example of a 3x3 NoC and the parts that constitute it. 52
Figure 2.19 - High-level overview of the NoC designs. 53
Figure 2.20 - NoC router with virtual channel strategy. 53
Figure 2.21- Examples of NoC topologies: (a) 2D mesh, (b) torus, (c) fat tree, (d) ring, (e) hypercube. 56
Figure 2.22 - Example of ideal irregular topology for an application specific. 57
Figure 2.23 - Examples of routing algorithms and their routes to send a message from node A to node B. 59
Figure 2.24 - Example of FIFOs in the input channels of a router. 62
Figure 3.1 - An example of HiNoC topology. 67
Figure 3.2 - (a) CrossRing, (b) CrossTorus, (c) MultiCross topologies. 69
Figure 3.3 - An example of HrNoC architecture. 69
Figure 3.4 - GigaNoC topology. 70
Figure 3.5 - (a) Concentrated mesh, (b) flattened butterfly and (c) hybrid topology. 71
Figure 3.6 - Power consumption of each part of the router. 74
Figure 3.7- (a) Area and (b) power consumption analyses for crossbars. (c) Power consumption for the parts of a crossbar. 75
Figure 3.8 - (a) Matrix using centralized crossbar where *Ps* denote processors and *Ms* memories. (b) bit slices (1-bit radix-8 crossbar) 76

Figure 3.9 - (a) Example of HiCIT architecture. (b) Scalability of the proposed topology is reached increasing the set of router + crossbar. 78
Figure 3.10 - HiCIT architecture composed of a crossbar switch on the local level and a mesh router in the global level. 79
Figure 3.11 – State machine to control the data transfer of the crossbar switch. 80
Figure 3.12 – Packet format of the HiCIT architecture. 82
Figure 3.13 - Two levels of HiCIT architecture and the bridges used to interconnect them. 83
Figure 3.14 - Example of the use of a Dual-clock FIFO for reading and writing from NoC and cluster with different operating frequency. (a) $Bridge_{RC}$, (b)$Bridge_{CR}$. 84
Figure 3.15 - ASHiNoC analyzer framework. 88
Figure 3.16 - Number of crossbar 2:1 MUXES according to the number of input ports. 90
Figure 3.17 - Critical path for each crossbar size defined to estimate the crossbar delay. 91
Figure 3.18 - TVOPD core graph. 94
Figure 3.19 - Example of HiCIT architectures varying the crossbar size for the same application. 95
Figure 3.20 - HiCIT interconnection average power for the best configuration, varying the number of routers on the global level. 96
Figure 3.21- HiCIT interconnection area and for the best configuration, varying the number of routers on the global level. 97
Figure 3.22 - HiCIT interconnection average Latency for the best configuration, varying the number of routers on the global level. 98
Figure 3.23 - The Optimal HiCIT topology for TVOPD with 9 clusters. 98
Figure 3.24 - Mesh and HiCIT topologies for NCS benchmark. 100
Figure 3.25 – Mesh and HiCIT topologies for TMPEG4 benchmark. 101
Source: elaborated by the author. 101
Figure 3.26 - Mesh topology for TVOPD benchmark. 102
Figure 3.27 - (a) Average power, (b) average network hops and (c) average latency for NCS, TVOPD and TMPEG4 in the HiCIT and mesh topologies. 105
Figure 3.28: (a) Area and (b) power consumption when the number of PEs increases. 107
Figure 3.29 - HiCIT power consumption divided in wires and cells. 108
Figure 3.30 - Percentage of SoC occupied by the interconnection topologies with the total area divided in wires and cells. 108
Figure 4.1 - HCS architecture. 113
Figure 4.2 - (a) VIP architecture, (b) pipeline stages in the packet-switching mode, (c) pipeline stages in the circuit-switching mode. 114
Figure 4.3 – ReNoC architecture router with the proposed multiplexers. 116
Figure 4.4 – Two possible configurations from ReNoC architecture. 116
Figure 4.5 – Use of switch box to compose the architecture presented by (MODARRESSI, 2011). 117
Figure 4.6 - (a) Latency and (b) Average power consumption considering the wire length for 65nm and 90nm. 119
Figure 4.7 - Latency and throughput results considering different switching possibilities for 600MHz (a), (b) and 1GHz (c), (d). 122
Figure 4.8 - Wire length adopted for the experiments for circuit switching mode. 123

Figure 4.9 - Three switching configuration possibilities: (a) packet switching, (b) unbuffered circuit switching and (c) buffered circuit switching. 123
Figure 4.10 - MINoC router architecture. 124
Figure 4.11 - Operation Mode Controller Architecture. 125
Figure 4.12 - Input Channel architecture. 126
Figure 4.13 - (a) Input buffer architecture and the (b) buffer and its pointers. 127
Figure 4.14 - Output Channel architecture. 128
Figure 4.15 - Example of MINoC configurations. 129
Figure 4.16 - Average latency results for MINoC in comparison with a conventional packet switching router for (a) TVOPD, (b) NCS and (c) TMPEG4 benchmarks. 132
Figure 5.1 - Ideal NoC proposal is obtained with hierarchy and adaptability. 136
Figure 5.2 – Mapping for HiCIT topology for TVOPD benchmark. 138
Figure 5.3 - Mapping for HiCIT topology for TMPEG4 benchmark. 138
Figure 5.4 – Mapping for HiCIT topology for NCS benchmark. 139
Figure 5.5 - Average latency results for HASIN in comparison with other strategies for (a) TVOPD, (b) NCS and (c) TMPEG4 benchmarks. 141
Figure 5.6 - Average latency results for HASIN, HiCIT and MINoC when the bandwidths are changed for (a) TVOPD, (b) NCS and (c) TMPEG4 benchmarks. 144
Figure 5.7 - Three random mappings for HiCIT topology for TMPEG4 benchmark. . 145
Figure 5.8 – Three random mappings for HiCIT topology for TVOPD benchmark.... 146
Figure 5.9 - Three random mappings for HiCIT topology for NCS benchmark. 146
Figure 5.10 - Results for HiCIT and HASIN topologies for random mappings for (a) NCS, (b) TMPEG4 and (c) TVOPD benchmarks. 148
Figure 5.11 - Core graph of D26 benchmark. 150
Figure 5.12 – Core graph of AV benchmark. 150
Figure 5.13 – (a) original mapping and (b) the mapping adapted for D26 benchmark from TMPEG4 topology. 151
Figure 5.14 - (a) original mapping and (b) the mapping adapted for AV benchmark from the D26 topology. 151
Figure 5.15 - (a) original mapping and (b) the mapping adapted for NCS benchmark from the TMPEG4 topology. 151
Figure 5.16 - Benchmarks mapped for topologies designed for other applications: (a) D26, (b) AV and (c) NCS. 153
Figure 5.17 - (a) Power Consumption and (b) area results for different interconnection strategies for the NCS and TVOPD benchmarks. 154
Figure 6.1 - Six points which led the ideal interconnection solution. 158
Figure 6.3 - Comparison between 2D and 3D NoCs for different benchmarks. 160
Figure 6.4 - Example of specific technologies for each layer of a 3D integration. 161
Figure 6.5 - (a) Memory hierarchy and (b) its disposition in a 3DIC. 161
Figure 6.6 - 3D floorplan for D26 benchmark. 162
Figure 6.7 - 3D floorplan for TVOPD benchmark. 162
Figure 6.8 - Number of TSVs x Yield. 164
Figure 6.9 - A comparison related to the number of pillars required for a (a) conventional 3D mesh topology and (b) the proposed hierarchical 3D topology. 166
Figure 6.10 - 3D-HiCIT architecture. 167
Figure 6.11 - 3D-HiCIT architecture. 167
Figure 6.12 - Example of (a) 3D-HiCIT and (b) 3D-mesh topologies for 32 PEs. 168
Figure 6.13 - TSV characteristics. 169
Figure 6.15 - Flow chart to estimate the power for the 3D designs. 170

Figure 6.16 - Area estimation comparing HiCIT and mesh topologies..... 173
Figure 6.17 - Percentage of chip area for each NoC topology. 174
Figure 6.18 - Percentage of chip area for wiring, switching and TSVs. 174
Figure 6.20 - Mapping for AV benchmark for (a) 3D-mesh and (b) 3D-HiCIT topologies..... 176
Figure 6.21 - Mapping for D26 benchmark for (a) 3D mesh and (b) 3D-HiCIT topologies..... 176
Figure 6.22 - Area results for D26 and AV benchmarks comparing mesh and HiCIT topologies..... 177
Figure 6.23 - Power Consumption for D26 and AV benchmarks comparing mesh and HiCIT topologies. 178
Figure 6.24 - Average latency comparisons considering mesh and HiCIT topologies for D26 benchmark..... 179
Figure 6.25 - Average latency comparisons considering mesh and HiCIT topologies for D26 benchmark..... 179
Figure 7.1 - Strategies approached in this work. 183

LIST OF TABLES

Table 1.1 - Current challenges presented in section 1.1 for interconnected designs of MPSoCs and the related solutions considered in this work. 30
Table 2.2 - Bus, crossbar and mesh NoC architectures. 63
Table 3.1 - Resume of the hierarchical proposals found in the literature. 72
Table 3.2 - Synthesis results for the synchronizer module considering two frequency possibilities. 84
Table 3.3 - Physical information of TVOPD cores. 93
Table 3.4 - Results for the best configuration for TVOPD benchmark. 99
Table 3.5 - Physical information of NCS cores. 103
Table 3.6 - Physical information of TMPEG4 cores. 104
Table 3.7 - Reduction in power, NoC hops and latency with the use of HiCIT when compared to a flat mesh topology. 106
Table 4.1- Comparison among the adaptive strategies related to the switching. 118
Table 4.2 - Synthesis results for MINoC proposal compared to a conventional router for NCS and TVOPD benchmarks. 134
Table 5.1 - TVOPD benchmark core descriptions. 139
Table 5.2 - TMPEG4 benchmark core descriptions. 140
Table 5.3 - NCS benchmark core descriptions. 140
Table 5.4 - AV benchmark core descriptions. 152
Table 5.5 - D26 benchmark core descriptions. 152
Table 5.6 - Power Consumption and Area reductions obtained for the NCS and TVOPD applications with HASIN strategy. 154
Table 5.7 - Latency parity for HiCIT operating at 1GHz and HASIN operating at 500MHz. 155
Table 6.1 - Topology, NoC size and number of cores defined in the experiments. 172
Table 6.2 - Physical information of AV cores. 177
Table 7.1 - Solutions for each challenge and the related publications. 186

ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
3DIC	Three Dimensional Integrated Circuit
AHB	Advanced High-performance Bus
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus
ASIC	Application-Specific Integrated Circuit
ASHiNoC	Application Specific Hierarchical NoC
AV	Audio Video
BCS	Buffered Circuit Switching
BIST	Built-In Self-Test
BNoC	Backbone NoC
bop	begin-of-packet
CMP	Chip Multi-Processor
CPU	Central Processing Unit
CS	Circuit Switching
Cu	Copper
DAMQ	Dynamically Allocated Multi-Queue
DBI	Direct Bond Interconnect
DPE	Data Processing Engines
DSE	Design Space Exploration
eop	end-of-packet
EVC	Express Virtual Channel
PE	Processing Element
FCFS	First-Come, First-Served
FF	Flip-Flop
FIFO	First-In First-Out
FPGA	Field-Programmable Gate Array
GALS	Globally Asynchronous Locally Synchronous
HASIN	Hierarchical Adaptive Switching Interconnection NoC
HiCIT	Hierarchical Crossbar-based Interconnection Topology
HCS	Hybrid Circuit Switching

IC	Integrated Circuit
ID	Identification Number
IEDM	International Electron Device Meeting
IL	Ideal Latency
IoC	In-order-Core
IP	Intellectual Property
ITRS	International Technology Roadmap for Semiconductors
LLC	Long Link Controller
LRS	Least Recently Served
MINoC	Multi Interconnections Network-on-Chip
MPEG	Motion Picture Experts Group
MPSoC	Multiprocessor SoC
MUX	Multiplexers
NI	Network Interface
NoC	Network-on-Chip
NCS	Network Communication Subsystem
OoC	Out-of-Order
PS	Packet Switching
QoS	Quality-of-Service
QPI	Quick Path Interconnect
RASoC	Router Architecture for System-on-Chip
RC	Routing Computation
ReNoC	Reconfigurable NoC
RISC	Reduced Instruction Set Computer
RLC	Resistance, Inductance, Capacitance
RR	Round Robin
SA	Switch Allocator
SAF	Store-and-Forward
SAFC	Statically Allocated, Fully Connected
SAMQ	Statically Allocated Multi-Queue
SoC	System-on-Chip
SoCIN	System-on-Chip Interconnection Network
TMEPG4	Triple MEPG4
TS	Topology Switch
TVOPD	Triple Video Objective Plane Decoder

UCS	Unbuffered Circuit Switching
UFRGS	Universidade Federal do Rio Grande do Sul
VA	Virtual Channel Arbiter
VC	Virtual Channel
VCT	Virtual Cut-Through
VHDL	VHSIC Hardware Description Language
VHSI	Very High Speed Integrated Circuit
VIP	Virtual Point-to-Point
VLIW	Very Long Instruction Word
VLSI	Very Large Scale Integration
WH	Wormhole
Xbar	Crossbar

TABLE OF CONTENTS

RESUMO **25**

ABSTRACT **27**

LIST OF FIGURES **29**

LIST OF TABLES **33**

ABBREVIATIONS **35**

1 INTRODUÇÃO **23**

1.1 Problem Description **24**
1.2 Motivation and Work Context **29**
1.3 Goals and Contributions **31**
1.4 Organization **32**

2 SYSTEMS, COMMUNICATIONS AND THEIR INTERCONNECTIONS **35**

2.1 Communication Characterization **35**
2.2 Before the NoCs **36**
2.2.1 Point-to-point interconnection **36**
2.2.2 Buses **38**
2.2.3 Crossbars **42**
2.3 The need for other alternatives to interconnect systems-on-chip **46**
2.3.1 The impact of wires on the interconnections **47**
2.4 Networks-on-Chip: interconnection model for complex systems **50**
2.4.1 Router **53**
2.4.2 Topology **54**
a) Direct Networks 55
b) Indirect Networks 56
c) Irregular Networks 56
d) Hierarchical Networks 57
2.4.3 Routing **58**
2.4.4 Switching **59**
2.4.5 Bufferization **61**
2.4.6 Arbitration **62**
2.4.7 Flow Control **63**
2.5 Comparison between the main interconnection devices **63**
2.6 Considerations **64**

3 2D HIERARCHICAL NETWORKS-ON-CHIP **65**

3.1 State of the art 2D hierarchical NoCs **66**
3.2 HiCIT topology **73**
3.2.1 Local level architecture **78**
3.2.2 Global level architecture **81**
3.2.3 Bridge architecture **81**
3.3 ASHiNoC tool **84**
3.3.1 Problem Formulation **87**
3.3.2 Overview of the application specific methodology **88**
3.4 Results **92**

3.4.1 ASHiNoC tool results 93
3.4.2 Synthesis results 106
3.5 Considerations 109

4 ADAPTIVE NETWORKS-ON-CHIP 111

4.1 State of the art adaptive switching NoCs 112
4.2 Switching Strategic analysis 118
4.3 MINoC architecture: an adaptive NoC 123
4.3.1 Operating Mode Controller 124
4.3.2 MINoC router input channel 126
4.3.3 MINoC router output channel 128
4.3.4 MINoC configurations 128
4.4 MINoC Results 130
4.4.1 MINoC Performance results 130
4.4.2 Synthesis Results 133
4.5 Considerations 134

5 HASIN – HIERARCHICAL AND ADAPTIVE NOC 135

5.1 Experimental Results 137
5.1.1 HASIN Performance Results 137
5.1.2 HASIN Performance Results when the application has a double bandwidth 142
5.1.3 HASIN Performance Results when the application has a random mapping 145
5.1.4 HASIN Performance Results when the application runs in a topology defined for other application 149
5.1.5 HASIN Synthesis Results 153
5.2 Considerations 155

6 THREE DIMENSIONAL NETWORKS-ON-CHIP 157

6.1 3D NoC paradigms 158
6.1.1 Reduction in the wire length and number of hops 159
6.1.2 Possibility to consider one layer for each technology 160
6.1.3 Layers with different number and size of cores 161
6.1.4 Heterogeneous interconnection 163
6.1.5 Interconnection with reduced number of TSVs 163
6.1.6 Hierarchical 3D NoC topology 165
6.2 Proposed Hierarchical 3D NoC 166
6.2.1 Hierarchical Architecture 166
6.2.2 Packet Format 167
6.3 Experimental Results 168
6.4 Considerations 179

7 CONCLUSION AND FUTURE WORKS 181

7.1 Major contributions 182
7.2 Open issues and future directions 187

REFERENCES 189

1 INTRODUÇÃO

Network-on-Chip, usually referred as NoC, has earned a space as an interconnection design due to its scalability and reusability in the multi and many core scenarios. As the number of processing elements (PEs) inside an integrated circuit has been increasing with technological advances, as well as system performance requirements, the interconnections of these systems has received a special attention. For up to a few processing elements, bus-based solutions may still be used (GUERRIER, 2000), (PANDE, 2005), (OETKEN, 2010), (SONG, 2009). However, as the number of components in these systems and their complexity scales up, the pressure on the bus system also increases. Therefore the use of bridged and multilayer buses becomes necessary, even though the protocol to control such a system gets more complex, presenting limited scalability, with variable and unpredictable performance. Moreover, at the physical level with technology scaling, while the gate delays decrease the global wire delays do not reduce in the same rate (ITRS, 2011), which will also impact the system performance. This issue surfaces in both general-purpose multicores such as Chip Multi-Processors (CMPs) and in heterogeneous application-specific Multi-Processor Systems on Chips (MPSoCs) (JERRAYA, 2005). They comprise of several components that need to be connected together in an efficient way, in order to sustain low energy consumption with enhanced performance.

Because of the aforementioned issues, over the last decade the use of Networks-on-Chip (NoCs) has been considered as an alternative to the interconnection infrastructure (TEWKSBURY, 1992), (GUERRIER, 2000), (HEMANI, 2000), (DALLY, 2001), (BENINI, 2002). The concept of switching networks originated in 1992 (TEWKSBURY, 1992), but only some years later did the use of a network-based interconnect begin to gain strength (GUERRIER, 2000). Finally the well-known NoC denomination was proposed by Hemani in 2000 (HEMANI, 2000). Therefore, SoCs (System-on-Chip) with NoCs have been a trend in the last years. Some examples of heterogeneous SoCs that use NoCs can be observed in the OMAP five platform (Texas) (TEXAS, 2013), Spear1310 and Spear1340 (ST, 2013). There are also some examples of homogeneous systems that use an NoC, such as the 48-core Intel Single-chip Cloud Computer (HELD, 2010) and the Tile64 (TILERA, 2013). This industry

movement means that old approaches such as point-to-point, crossbars, bus and others can no longer meet the communication requirements for some of the current systems.

One of the advantages of using NoCs as interconnection infrastructure is its modularity (DALLY, 2001), (PANDE, 2005), (BOLOTIN, 2004), (PASRICHA, 2008). It is a key asset in supporting scalability from the ground up, particularly in terms of performance. Physical-design-aware NoC components enable large-scale System-on-Chip designs, with more predictable (and possibly guaranteed) performance (BENINI, 2002) (NICOPOULOS, 2010). Moreover, the distributed nature of NoC infrastructures with redundancy of components can be effectively manipulated to enhance the system-level reliability (DE MICHELI, 2010).

An NoC can be described mainly by its topology, the strategies used for routing, flow control, switching, arbitration and buffering. The network topology can be represented as an arrangement of nodes and channels on a graph. The routing strategy determines how a message chooses a path through the topology, while the flow control deals with the allocation of channels and buffers of a message, as it follows the chosen path. The switching mechanism removes data from a router input channel and places it in an output channel, while the arbitration strategy is responsible for scheduling the use of channels and buffers by the messages (ZEFERINO, 2003). A switching network presents a set of switching elements with physical connections between their terminals (GUERRIER, 2000). NoCs emerged with the objective of transmitting the messages in packets, moving data from one node to another. The basic component of a NoC is the router whose function is to forward the incoming packets to one of its neighboring routers. This process is repeated in each router until the packets reach the final destination (GUERRIER, 2000).

In the next section it will be discussed the problems foreseen for the next years in terms of interconnecting designs for complex systems. Following that, the work context and the adopted solutions for each related problem are presented in section 1.2, identifying the reasons that point out the need for other alternatives of networks-on-chip. The goals and contribution of this work are described in section 1.3 and finally, the document organization is discussed in section 1.4.

1.1 Problem Description

The reduction in nanoscale technology and consequently, in device sizes has led to a considerable increase in the chip density (WEERASEKERA, 2009), (PASRICHA, 2009),

(ITRS, 2011). Therefore, in association with this evolution, interconnection design for these systems is not so trivial, since the number of processing elements (PEs) in a system is continuously growing and the systems are becoming increasingly complex (NICOPOULOS, 2010), (KUMAR, 2012).

Until a few years ago, PEs were interconnected by buses, crossbars or ad-hoc interconnection structures. However, due to the limitations of these interconnection devices, new interconnection designs present other features that should be considered, such as:

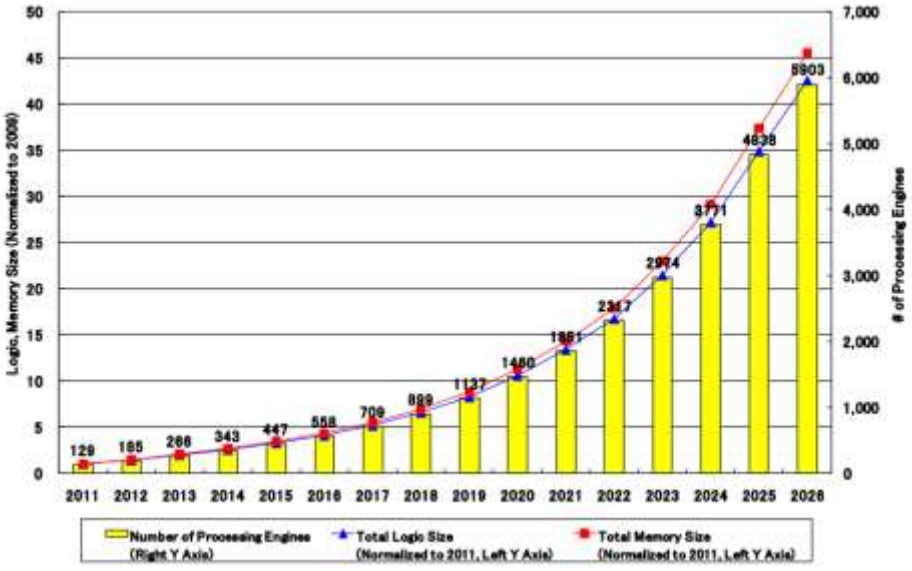
- System heterogeneity;
- Requirements for high performance associated with awarded power consumption;
- Need to allow the system to be updated;
- Possibility of executing different applications considering the same interconnection device.

Then, due to all these requirements and huge design space, intelligent networks-on-chip, with features applied for different scenarios, need to be applied. An NoC can have many features tuned to provide high performance with low overheads for each application. In an NoC, the link-level, as well as the router design and protocol level, can all be configured to achieve the specifications of each application. The goal can be related to the Quality-of-Service (QoS), physical constraints and/or reliability issues, and each system can have goals in specific aspects. As a result of these questions, the first NoC architectures no longer meet the requirements of the current and next MPSoC designs. Several proposed NoC architectures for NoCs with the purpose of reducing the energy consumption and increasing the system performance can be found in the literature. Nevertheless, several of these proposals present very expensive architectures, composed of virtual channels, tables and complex controls. With the addition of all such hardware resources, these NoC architectures can be prohibitive in the field of embedded systems. In this manner, the problem related to the system interconnection is not satisfactorily solved and so other alternatives need to be proposed. This work focuses exactly on proposing different techniques for varied scenarios of MPSoCs. In order to understand the future MPSoCs needs, it is important to highlight the main NoC challenges, which are as follows:

1. Large latency in the transmission of messages due to a high number of hops between the source and destination, since the trend for SoCs is a curved increase in the number of PEs (ITRS, 2011), (PASRICHA, 2009), (WEERASEKERA, 2009). Hops is the average number of switches that a

packet traverses to reach the destination node (PAVLIDIS, 2011). Figure 1.1 was extracted from International Technology Roadmap for Semiconductors (ITRS, 2011) and shows the trend in the number of Processing Engines. Processing Engines are processors customized for a specific function. A function with a large-scale, highly complicated structure which will be implemented as a set of PEs. This architecture template enables both high processing performance and low power consumption, by virtue of parallel processing and hardware realization of specific functions. The architecture does not require specific processor array architectures or symmetric processors; its essential feature is its large number of PEs embedded within the SoC to implement a set of required functions (ITRS, 2011). Thus, SoCs will potentially have a large number of PEs in order to achieve the required performance objectives.

Figure 1.1 - Trend f number of PEs for SoC Consumer Portable Design Complexity.



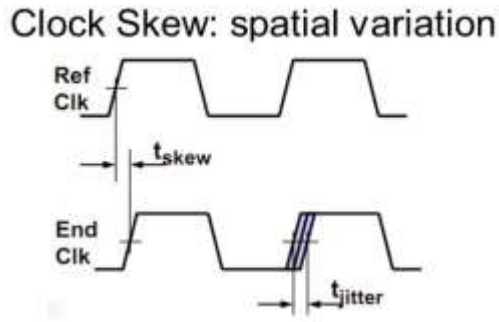
Source: ITRS (2011, p.7).

2. Need to obtain NoC designs with a very high performance associated with low power and energy consumption. Many proposals present an elevated increase in power consumption to reach high performance (JERGER, 2008), (MODARRESSI, 2010). However, in the embedded systems context, the power features also need to be taken into consideration (DAS, 2009), (CHOU, 2010), (ITRS, 2011).

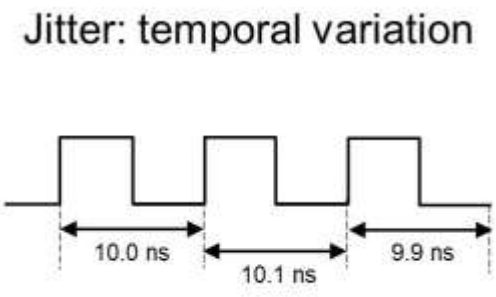
3. Very complex systems introduce long wires in the distribution of clock, causing clock skew and jitter (BEIGNÉ, 2006), (SHEIBANYRAD, 2007), (PANADES, 2007), (THONNART, 2009), (STRANO, 2011). Clock skew is defined as a spatial variation of the clock signal. The clock skew is measured by the difference between the reference point and a particular destination point. In this case, the clock signal can arrive at different points at different times (OKLOBDZIJA, 2003). Clock jitter is defined as temporal variation of the clock signal relative to the reference transition (reference edge) of the clock signal (OKLOBDZIJA, 2003). Figure 1.2 illustrates how these two situations occur.

Figure 1.2 - (a) Example of Clock Skew and (b) Jitter.

(a)



(b)

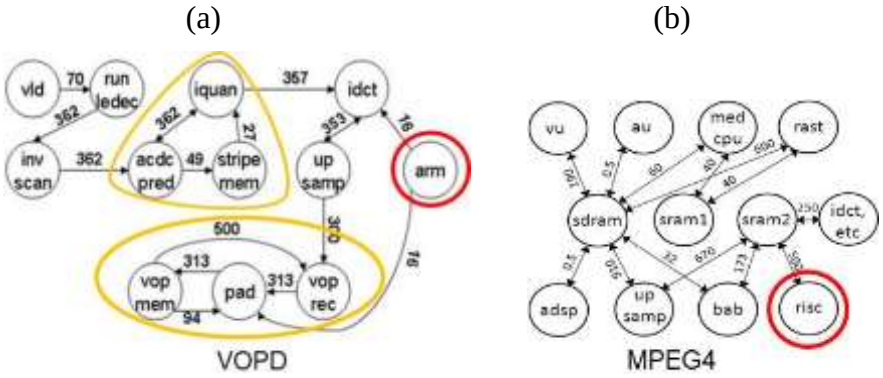


Source: elaborated by the author.

4. Need to meet different requisites and communication rates to provide the required performance at runtime, even when there are changes in the bandwidths of the cores (NICOPOULOS, 2006), (MATOS, 2011c), (BOBDA, 2005), (YING-CHERNG, 2011). In order to better understand this requirement, Figure 1.3 presents two very common benchmarks (BERTOZZI, 2005). Analyzing these figures, the bandwidth requisites of each communication can be observed. In addition, the same core used in other

applications has a different function, as, for example a RISC processor presents a specific role in each benchmark. In this manner, the same system must be able to run different applications and have its bandwidths updated according to the application being executed. In such a case, modern MPSoCs present a high complexity and they must efficiently handle some situations not foreseen in the design time.

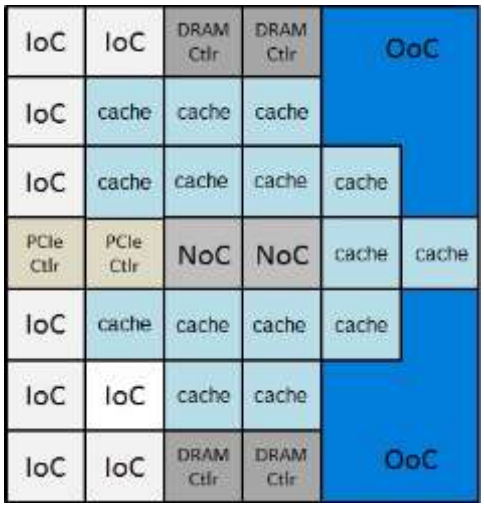
Figure 1.3 – (a) VOPD and (b) MPEG benchmarks with the communication core graph.



Source: BERTOZZI (2005).

5. Systems composed of many applications, presenting different communication needs, such as a large variation in the communicate rates due to the heterogeneity of the cores (HU, 2006), (STENSGAARD, 2008). This point is also evidenced in the figure 1.3. A common observation related to these benchmarks is the heterogeneous communication nature of SoCs, as each core only communicates with few others with a specific rate. These features can be found out from the current systems in the industry, like Tile-Gx (TILERA), OMAP5430 (TI), SPEAr1310 (ST). These systems integrate cores with different functions and requirements. The concept of Out-of-Order cores (OoCs) and In-order-Cores (IoCs) were introduced in (MANFERDELLI, 2008), where distinct communication behaviors are demanded in an MPSoC. OoCs are larger and dissipate more power than IoCs. Moreover, there is more communication among IoCs than among OoCs. Figure 1.4 exemplifies an MPSoC composed of OoCs and IoCs. An adequate NoC design must be able to manage these different requisites of an MPSoC.

Figure 1.4 - Example of cores in an MPSoC.



Source: MANFERDELLI (2008).

6. Increase in the wire delay and wire power caused by the reduction of the technological scale (PARSRICHA, 2009), (WEERASEKERA, 2009), (ATIENZA, 2008). The wire does not scale with the gates due to the increment of wire resistance (HO, 2001). This characteristic will be better explained in subsection 2.3.1.

Based on the main problems related in this section, it was possible to observe the real requirements to be introduced in the proposed solution. In the next section, the strategies to solve these problems will be discussed.

1.2 Motivation and Work Context

When this work began, there was a lot of research related to networks-on-chip but little use in the industry. Nowadays, the use of a network-on-chip in the industry is a reality and necessity. Besides it is also possible to find specific companies designing NoCs, (ARTERIS, INOCs, SONICS, BLENDICS), confirming that old interconnection devices do not comply with the current design requirements.

This work considers several essential features for new SoC designs, focusing on what the ideal interconnection solution for these systems would be. The studies and proposals raised in this work take into account the NoC topologies, adaptabilities, switching mechanisms and the new context of design in 3 dimensions and their interconnections. From these analyses it is clear that conventional NoC solutions, as for example, a 2D mesh topology

with a simple packet switching mechanism, without any possibility of adaptability, do not meet the requirements for the future systems.

The proposal of this thesis presents combined solutions for the aforementioned challenges. Table 1.1 correlates the appropriate solution for each challenge. As can be observed from this table, it is possible to obtain an interconnection solution that meets all the raised issues with just four appropriately employed strategies.

Table 1.1 - Current challenges presented in section 1.1 for interconnected designs of MPSoCs and the related solutions considered in this work.

Challenges	Solutions	Chapters
1 Large systems increase the system latency	✓ Hierarchical interconnections ✓ Adaptive strategy ✓ 3D NoCs	3, 4, 6
2 High performance associated with low power	✓ GALS ✓ 3D NoCs ✓ Hierarchical interconnections ✓ Adaptive strategy + hierarchical topology	3, 5, 6
3 Clock skew and jitter	✓ GALS	3
4 Need to meet different system requirements	✓ Adaptive strategy	4, 5
5 Heterogeneous Systems	✓ Hierarchical interconnections	3
6 Increase in the wire delay and wire power	✓ 3D NoCs	6

The problem related to the size and complexity of the current and future systems (challenge 1) can be solved using a topology able to reduce the communication latency, like hierarchical and 3D topologies (GUERRE, 2010), (KUMAR, 2012).

As this work considers heterogeneous systems, the topology exploration is also one manner to combine high performance with low power (challenge 2 and challenge 5) (HOLLTEIN, 2006). Another ideal alternative for heterogeneous systems is to define the operating frequency according to the requirements of each part of the system. This can be provided with the use of operating frequencies defined for system isles, avoiding specifying a higher operating frequency than necessary for parts of the circuit, which would result in larger power consumption (LUDOVICI, 2011). The technique that allows the use of different operating frequency is called GALS (Globally Asynchronous Locally Synchronous). In GALS systems, the design can be partitioned in different frequency islands and the interconnection infrastructure is isolated by dual clock FIFOs at the boundary (LUDOVICI, 2011). The use of GALS is also a solution for problems related to clock skew and jitter (challenge 3), since with GALS the problems of clock distribution in the chip are reduced (BEIGNÉ, 2006). Still regarding challenge 2, the use of an adaptive strategy for NoCs is always an alternative for this problem. The adaptive NoC solutions can still be applied when the system or application has alterations or updates in their communication pattern (challenge 4) (FARUQUE, 2008), (NICOPOULOS, 2006).

An efficient solution related to the long links generated due to the large number of processing elements in a system, is the three dimensional (3D) designs. 3D designs allow shorter wires inside each chip layer and also short vertical interconnections between the layers (challenge 6) (SHEIBANYRAD, 2010).

1.3 Goals and Contributions

The main goal of this work is to provide a set of solutions for the next generation of heterogeneous MPSoCs, which are able to obtain high performance combined with low power consumption and small chip area. In order to accomplish this objective, a novel hierarchical topology and an intelligent adaptive switching mechanism for NoCs were developed. The hierarchical proposal is a combination of mesh topology and crossbars. These strategies were also explored in network-on-chip in three dimensional. The need for each proposed solution is linked according to the:

- system complexity;

- number and kind of cores;
- bandwidth;
- power requirements;
- need to update the application.

For example, if it is a specific application that demands high performance but would not present changes in the communication traffic, the ideal interconnection solution in terms of topology from an algorithm that takes into account the application behavior, can be specifically defined for that system. However, if the system is large and composed of heterogeneous components designed from different technologies; the solution could be the use of a hierarchical 3D NoC.

Solutions for 2D designs, such as, hierarchical topology and adaptive strategy, will be presented. The integration of these two solutions in a same NoC is investigated, analyzing the situations where advantages in this employment are obtained. Besides this, some experiments for 3D designs are also started in this work, expanding some of the 2D solutions to the 3D context. Results for different scenarios give evidence of the advantages of the proposed solutions compared to others.

The major contribution of this work is to succeed in the use of a simple topology architecture that provides small area and power consumption with high performance due to the hierarchy where the communication locality must be explored. Combined with this solution, when the initial topology needs to meet with variations in the system communication pattern, an adaptive solution allows low latency to be ensured even in intra-cluster communications. Different studies were analyzed to verify the effectiveness of this solution, such as using a specific hierarchical topology to run different heterogeneous applications and increase the bandwidths in the inter clusters. As well as this, it was proven that the proposed topology match very well with the requisites of a 3D design, as it allows the reduction of the number of vertical interconnections, which are expensive and prone to defect. Results show even more benefits in the use of the proposed topology in three dimensional.

1.4 Organization

This thesis is organized into 7 chapters. Before presenting the contributions, Chapter 2 provides an appropriate background of the work in this thesis. As the hierarchical proposal presents a hybrid NoC, different alternatives already used, such as buses and crossbars, are discussed as well as their advantages and disadvantages compared to current solutions. The

need and reasons that led to other interconnection solutions like the networks-on-chip are also discussed in this chapter. Following that, the main NoC features are presented.

Chapter 3 introduces the motivation and state-of-the-art hierarchical topologies. Next, the proposed hierarchical topology and its architecture are presented. An automatic tool was developed to find an optimal mapping for the specific proposed topology. From this tool, some initial results are discussed for some benchmarks.

Chapter 4 presents another proposal for NoCs based on adaptability. This proposal explores switching modes for networks-on-chip. A strategy that allows using up to three switching modes was investigated. Firstly, the state-of-the-art and related works about switching adaptability in NoCs are discussed. After, the proposed adaptive NoC architecture called MINoC is presented. In the end of this chapter, performance and synthesis results are discussed.

Chapter 5 extends the integration of the two previous proposals in a same NoC. This strategy was adopted to be able to manage with possible changes in the communication rates foreseen at design time. In this manner, the high performance is ensured even when the system is updated or suffers alternations in the bandwidths of cores. Different situation were evaluated in order to prove the resilience of the proposed adaptive and hierarchical solution.

Chapter 6 presents a recent need in interconnection: network-on-chip for 3D designs. In this chapter a set of 3D design features and many factors that impact directly in an interconnection solution are raised. From these points of view, the advantages obtained when one makes use of the proposed hierarchical topology in 3D integrated circuits are perceived. Results for the previous strategies are presented and compared with other solutions.

Finally, chapter 7 provides concluding remarks on the work presented. The chapter summarizes the thesis, outlines its contributions and proposes future research directions.

2 SYSTEMS, COMMUNICATIONS AND THEIR INTERCONNECTIONS

2.1 Communication Characterization

Before presenting the features of the systems and their interconnection devices, it is important to define some properties of these components to later use them to compare architectures.

Properties:

- **Latency**: Latency refers to how long the data/message takes from source to destination. It is indicated by the absolute time for the completion of a task (KOGEL, 2006). A latency guarantee occurs whenever a data unit crosses the communication architecture and reaches its destination within a finite amount of time (PASRICHA, 2008). It is usually measured as a time unit, for example seconds or cycles.

- **Throughput**: in general refers to the amount of information over time. This information can be defined on several levels of granularity, for example, tasks, data, bytes, bits and still in packages or flits in the NoC context (PASRISHA, 2008).

- **Bandwidth**: refers to how much data can be moved per time unit. A bandwidth guarantee concludes that a group of data units passes through a portion of the communication architecture at a certain data rate (PASRISHA, 2008).

- **Power Dissipation**: this is a measurement that denotes the energy per time. In order to better understand this measurement, one considers two situations. For example, the battery lifetime of a smartphone; it depends directly on the energy consumption; however the packaging costs depend on the heat dissipation properties which are measured by the power consumption (KOGEL, 2006).

Computational Efficiency: efficiency is derived from performance and power consumption (bits/s x W). This measurement is very relevant since it characterizes the efficiency of a given architectural element taking into account the costs of obtaining a required performance. It is easy to obtain architecture with high performance but consuming a lot of power. In this case, the efficiency will be low.

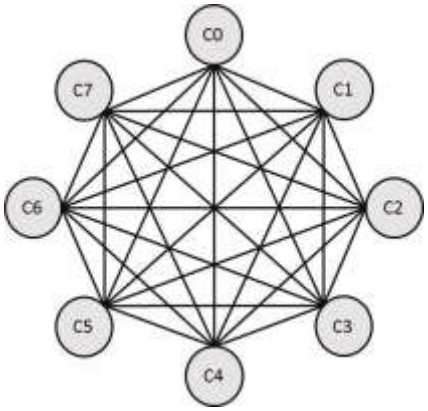
2.2 Before the NoCs

As this work will present and propose some hybrid interconnection devices, it is important to understand the evolution and the reasons for that. Besides, it is interesting to analyze when it is more appropriate to use one solution in place of another. Thus, following the main interconnection architectures before the introduction of NoCs will be verified.

2.2.1 Point-to-point interconnection

The simpler possibility to interconnect the devices of a system is point-to-point (Figure 2.1). In this case, each device is directly connected to another by a dedicated path. Although this alternative can present high performance, since the message can be sent with a minimal delay, this option is inflexible and presents limited scalability. This happens because according to the number of core increases, the physical implementation becomes more difficult due to the routing and the long wires. Analyzing the floorplan of a large system interconnected by point-to-point, long wires are required to cross the core areas.

Figure 2.1 - Example of point-to-point interconnection.



Source: elaborated by the author.

In industry, it is possible to find some examples that use point-to-point concepts. Intel proposed the QuickPath (QUICKPATH, 2008) to interconnect multicores. The differential of this device relative to other older solutions was that it presented one path to send data and another to receive. Besides this, QuickPath Interconnect (QPI) presents a path separated for memory and another for devices. One of the biggest changes compared to other alternatives

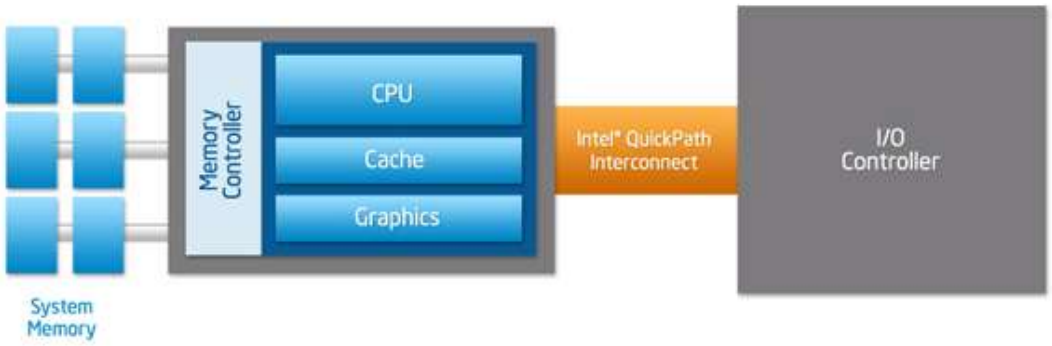
used by Intel, is the implementation of scalable, shared memory, since old devices use a single shared pool of memory connected to all the processors by a bus. With QPI, each processor has its own dedicated memory that is accessed directly through an Integrated Memory Controller. So, whenever a processor needs to access a dedicated memory of another processor, it can do through QPI where all processors are linked.

Another example of point-to-point interconnection, designed to increase the communication speed is HyperTransport at AMD (HYPERTRANSPORT, 2003). HyperTransport was designed five years before QuickPath and both present similar speeds, although these two devices are completely incompatible. The interconnection used by AMD separates data paths for input and output, enabling the processor to read and write at the same time. QuickPath also employs double data rate technology to squeeze two data transfer per clock. It has a variable bit-width between 2 and 32 bits.

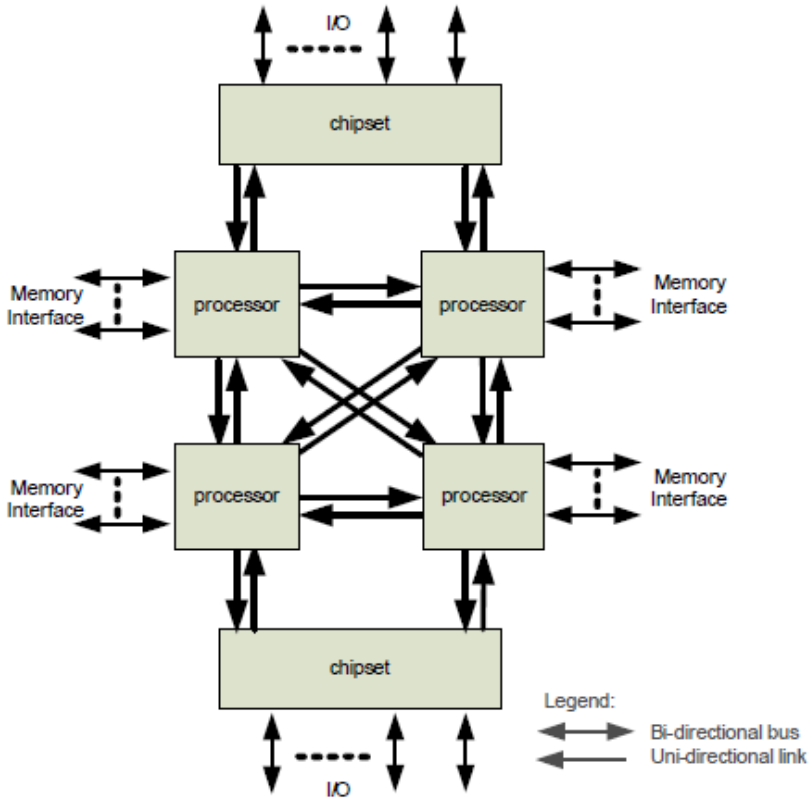
With these two examples, an inverse process has occurred since these interconnection devices emerged to replace the buses used previously. In this case, this change was defined due to the high speed of the point-to-point compared to the buses. As these systems are composed of only a few cores (Figure 2.2(b)), this alternative complies with the design requirements.

Figure 2.2 - (a) Example of system interconnected by Intel QuickPath and (b) its architecture.

(a)



(b)



Source: INTEL (2008).

2.2.2 Buses

Buses are the most common devices for system interconnection and they have been largely used in the last decades by the industry, still being employed in many designs nowadays. In fact, depending on the system, it can be an optimal interconnection solution and

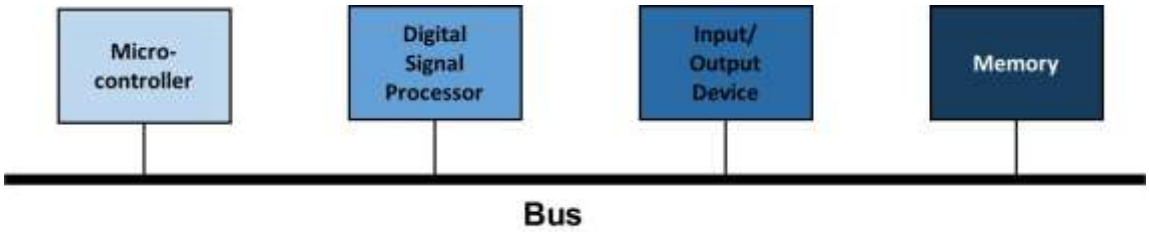
this definition is decided in function of the core communication rates, the bus architecture and the system size.

The most notable feature of a bus is its simplicity. The basic function of a shared bus is to transfer a single message at a given time. All IPs (Intellectual Properties) are interconnected in the same device and, therefore, only one IP can access the bus, while the others wait their time. However, because of the way they are shared, buses are very limited and present a control based on masters and slaves. IPs defined as masters initiate a read or write data transfer. IP slaves never can initiate a transfer; they only reply to the transfer requests. However, as many masters can request the bus simultaneously, an arbiter is necessary. The arbiter monitors the requests and considers an algorithm to choose the master with the highest priority request. A bus also requires a decoder to specify to which component the transfer corresponds. The decoder block generates the select lines that define the target slave. The transaction address is decoded and from this, it is verified which slave base address stored in the registers, it matches. Buses can be a good option for broadcasting with a few IPs, since it is simple to implement, albeit that they present some disadvantages, such as: limited performance, poor scalability, limited frequency and others.

The most simple bus topology found in many SoC architectures is the Shared Bus, as depicted in Figure 2.3. In this model, several masters and slaves are connected in the same structure. A bus arbiter examines the requests from the master interfaces and grants access according to the specified bus protocol. In this topology, the bus load is increased adding masters or slaves, limiting the bus bandwidth.

Besides the implementation simplicity, the advantage of this topology is the low area costs of few cores. The disadvantages of shared-bus architecture are large load per data bus line, long delay for data transfer, large energy consumption, and low bandwidth (MITIC, 2007). These disadvantages occur since every additional IP block connected to the bus adds to this parasitic capacitance, in turn causing increased propagation delay. As the number of IP blocks increases in a bus, the associated delay in bit transfer over the bus may grow and will eventually exceed the targeted clock period (PANDE, 2005).

Figure 2.3 - Example of a system interconnected by a shared bus.



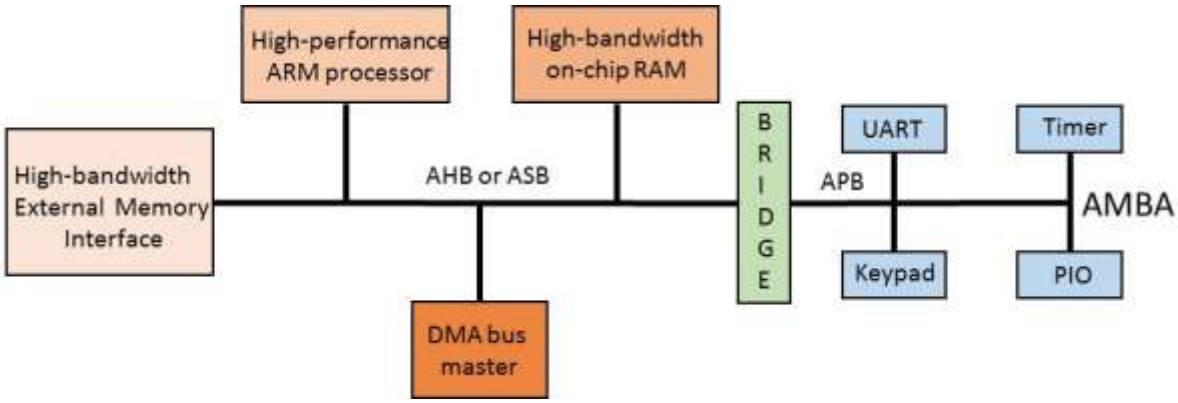
Source: elaborated by the author.

There are several bus architectures largely used in the industry, some examples are AMBA (Advanced Microcontroller Bus Architecture) Bus by ARM (AMBA, 2013), CoreConnect by IBM (IBM, 2007), Wishbone by Silicore Corporation, Avalon by Altera (SHARMA, 2012). The AMBA architecture is largely employed. The general view of the AMBA blocks is presented in Figure 2.4.

However, bus-based interconnections have performance bottlenecks as the number of cores increases (GUERRIER, 2000). Long buses (with many components) become slow due to large RC (resistive-capacitive) delays derived from each attached core (SILVANO, 2011, Chapter 1). Because of this, in order to increase the bus throughput, hierarchical buses have been considered. Hierarchical bus architecture consists of several shared buses interconnected by bridges. In this case, the bridges are responsible of interconnecting the buses, acting as a slave from one side and a master from the other. Cores are placed at the level in the hierarchy according to the required performance. However, transactions across the bridges involve additional overheads and during the transfer both buses remain inaccessible to the other components (MITIC, 2007). The increase in throughput is obtained as a result of the reduced load per bus and the possibility of having parallel transactions on different buses.

AMBA has a hierarchical bus structure and can be composed of two different bus features in each hierarchical bus. For example, AHB and ASB are used for high performance, allowing pipeline operation for multiple masters (Fig. 2.4). AHB is also used for burst transfers and split transactions. However, the APB is proposed for low power, with a simpler interface (Fig. 2.4) (AMBA, 2013). Several masters and slaves are connected to AHB and an arbiter decides which master will have data transferred. Data is transferred in bursts that involve the reading/writing of addresses and AHB is connected to APB via a bus bridge (ROYCHOUDHURY, 2003).

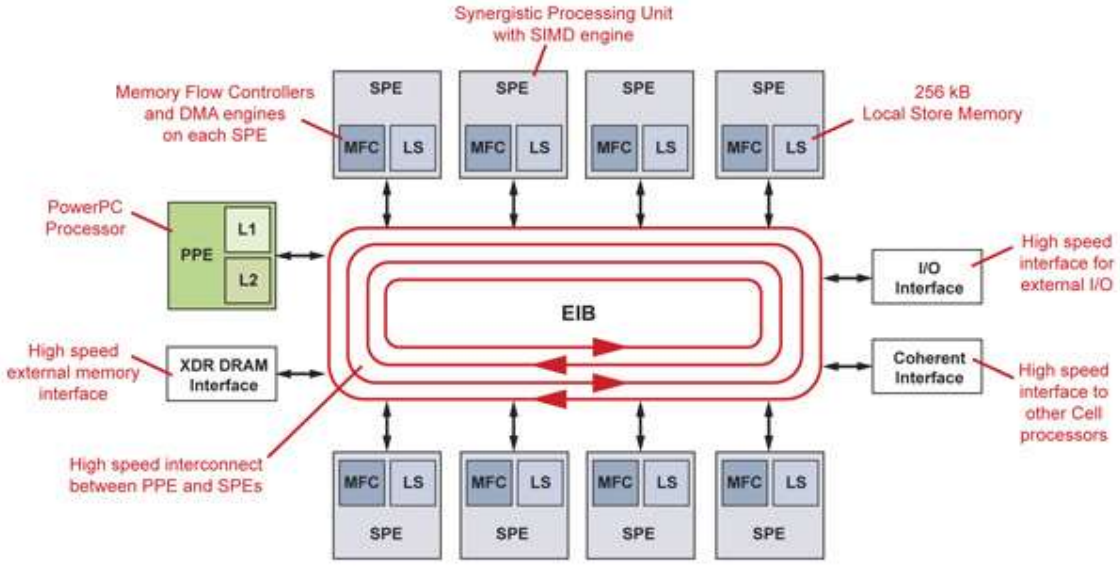
Figure 2.4 - Hierarchical AMBA bus architecture.



Source: AMBA (2013)

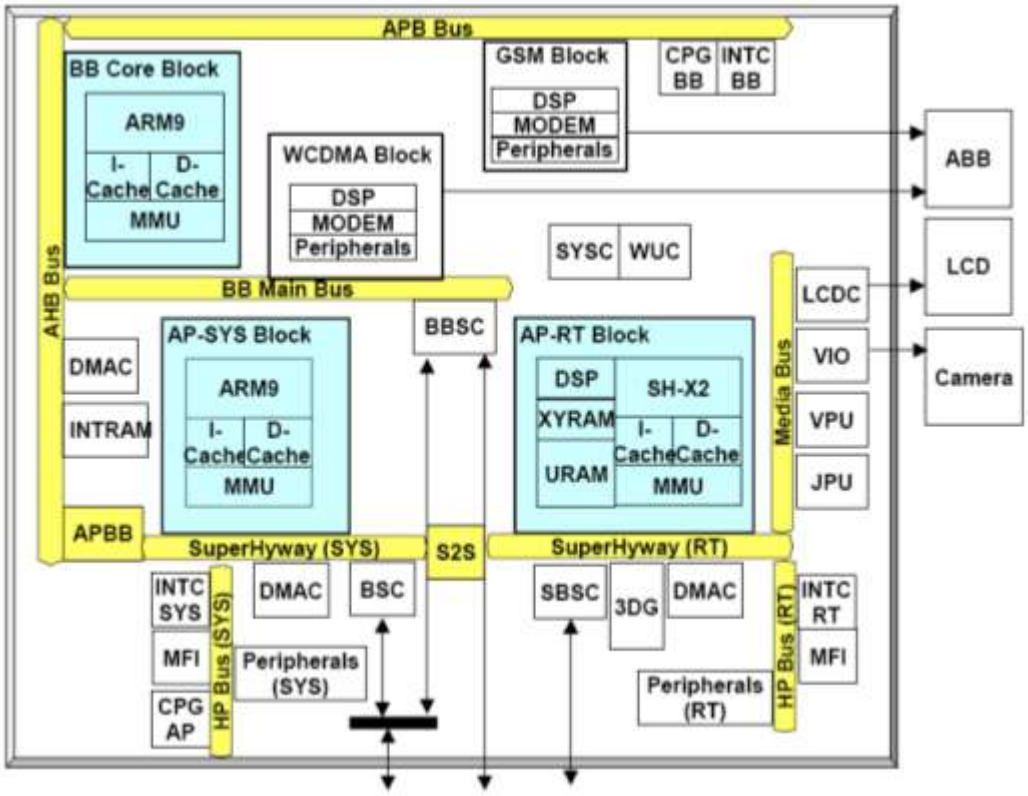
Other bus variations have been implemented by the industry in order to increase the throughput and reduce latency. Examples of this are depicted in Figures 2.5 and 2.6. In the first case, it is the IBM cell, a heterogeneous multicore processor (ROSTRUP, 2010) that uses a ring bus to interconnect the cores (PowerPC, synergistic Processing Elements and interfaces). The second figure (Figure 2.6) is the SH Mobile architecture (RENESAS, 2013) where the components are interconnected by a hierarchical bus, each with a specific bus feature.

Figure 2.5 - IBM Cell ring bus communication architecture.



Source: ROSTRUP (2010).

Figure 2.6 – SH Mobile G1 hierarchical bus communication architecture.



Source: RENESAS (2013).

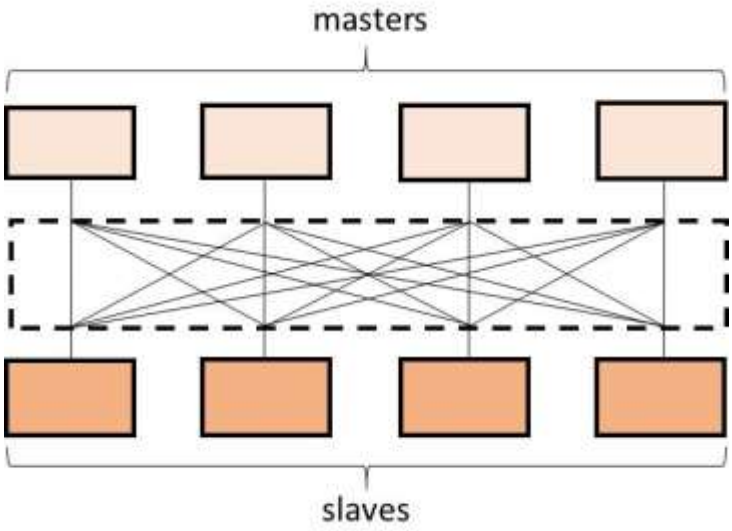
The wide variety of buses presents the problem for reusable design, even though a number of companies and standard committees have attempted to standardize them (MITIC, 2007).

Although nowadays there are more efficient interconnection devices for System-on-Chips, the buses are still largely used in the industry and satisfactorily meet their system requirements, since these systems are not very complex in terms of bandwidth and number of IPs.

2.2.3 Crossbars

The crossbars appeared as an alternative to buses. Crossbars, also called bus matrix or bus crossbar (Figure 2.7), are simple components where the cores are interconnected by a switch. An arbiter is considered whenever there is competition for the same destination. In the simpler model, all cores can send messages to all others and, if the number of masters is equal to n, n transfers can be made in parallel. This possibility is the main advantage of the crossbars compared to the buses, since it allows concurrent transfers, improving the performance. Nevertheless, crossbars use more wires and logic components (one arbiter and multiplexer for each output port) which result in larger power consumption and area overheads (PASRICHA, 2008).

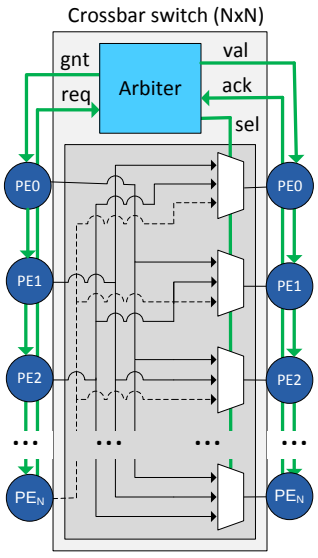
Figure 2.7 - Example of a full bus crossbar.



Source: elaborated by the author.

In the example of Figure 2.7, a full bus crossbar is presented. The internal crossbar architecture is presented in Figure 2.8. For each output port, one arbiter and one multiplexer are required, which can be verified. Other options considering partial bus crossbar have been proposed whenever the system does not require communication among all components. However, partial crossbars are less flexible, albeit that they reduce the number of wires. An example of commercial SoC with a full bus crossbar topology is the Niagara from SUN. This architecture connects eight SPARC processors, each one supporting up to four threads. Several efforts have shown the advantages of a full bus crossbar to significantly increase the throughput compared to the hierarchical bus strategies (PASRICHA, 2008).

Figure 2.8 - Crossbar architecture: wires, multiplexers and arbiters.



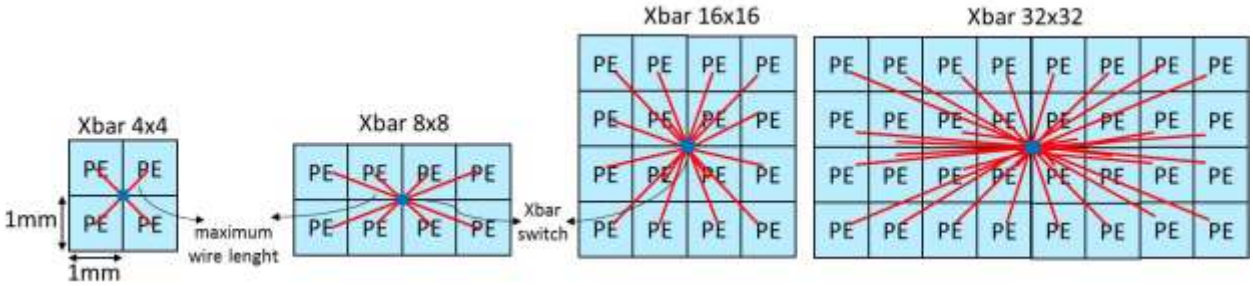
Source: elaborated by the author.

As one of the proposed solutions of this work considers the use of a crossbar switch in the topology, some relevant analysis will be demonstrated in this chapter. One aspect much discussed is the use of crossbars concerning scalability. This occurs due to the long wires required as the number of cores increases. Long wires damage the latency communication and increase power. In order to verify the latency of a crossbar, the maximum wire length required in each configuration was considered. This evaluation was obtained for 65 nanometers process technology from Hspice tool, using the RLC-π model (SAKURAI, 1983) describing electrical wire behavior. As the size of each core directly impacts the wire length, they were considered in these experiments, assuming interconnections for cores with areas around 1mm², similar to the area of ARM processors (ARM, 2013) for the same technology. Figure 2.9 presents a simple example of floorplanning, assuming that the crossbar switch is in the middle to provide the interconnections and that the cores are homogeneous.

This experiment is only an estimation to understand the crossbar limitation. The reference of the crossbar size is related to the number of input ports X output ports. For example, a full crossbar 4x4 means that it can connect 4 cores and each core will have 4 input and 4 output ports. For this analysis homogeneous cores were considered. In this manner, the maximum latency was obtained from the longest wire required to interconnect two cores through of a crossbar. As the crossbar does not have any mechanism of restoring the signal, in the simulations a repeater after every 2mm of wire length was included. Figure 2.10 shows the latency results for the wires and logic components required for each case. This latency was calculated from the sum of the wire and crossbar delay. The crossbar delay was obtained from Hspice simulations. In this case, the set of multiplexers required for each crossbar size were implemented. As can be observed in Figure 2.10, for this experiment crossbars interconnecting around 16 cores do not compromise the performance, in terms of wire latency. A crossbar to interconnect 16 cores still allows the definition of a high frequency for the circuit (around 700MHz). However, it can be seen that the wire latency is much larger for crossbars with 32 ports interconnecting cores with an area equals 1mm².

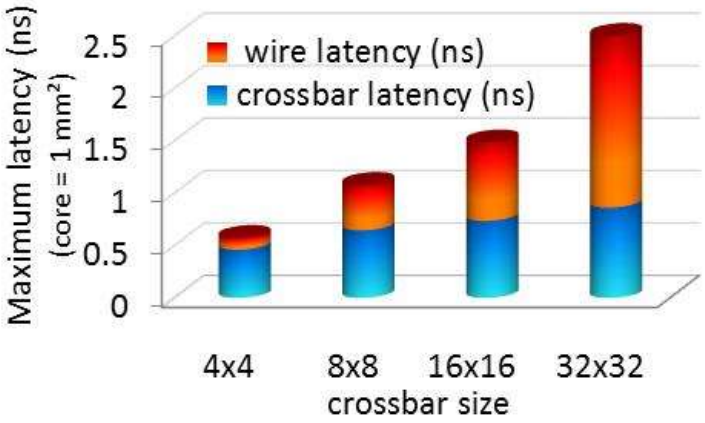
It is clear that these results are totally dependent on the core areas. For example, if the same analysis is done for an ARM946E-S processor (ARM, 2013) with an area equals 0.488mm² (processor + cache) for TSMC65LP technology process, larger crossbars can be used. The results for these core areas are presented in Figure 2.11. As previously stated, the core areas directly affect the maximum operating frequency. For this new analysis, crossbar circuits with 16 ports can operate with more than 1GHz and crossbar circuits with 32 ports, almost 800MHz.

Figure 2.9 - Example of maximum wire length according to the number of PEs interconnected by a crossbar.



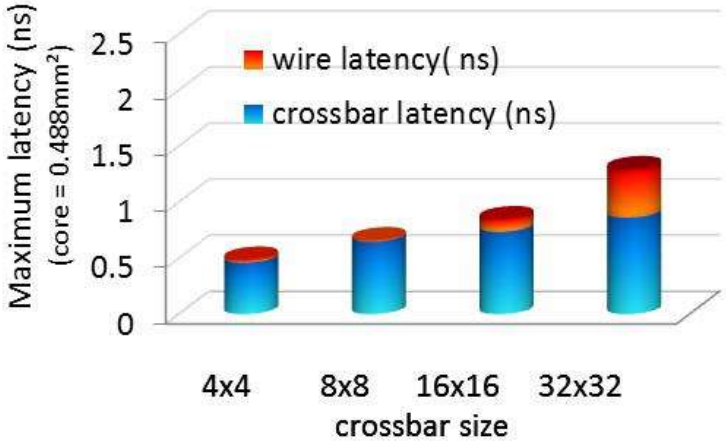
Source: elaborated by the author.

Figure 2.10 - Maximum latency according to the crossbar size considering cores with an area of 1mm².



Source: elaborated by the author.

Figure 2.11 - Maximum latency according to the crossbar size considering cores with an area of 0.488mm².

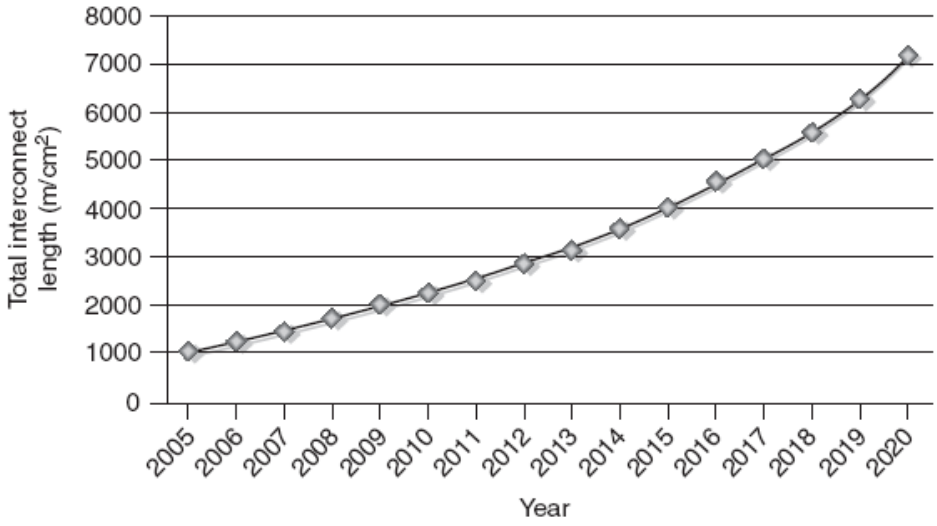


Source: elaborated by the author.

2.3 The need for other alternatives to interconnect systems-on-chip

The emergence of more complex interconnections succeeds from the scaling of MOS transistors into the nanometer regime associated with the increase in the microelectronic system complexity and the advent of innovative VLSI (Very Large Scale Integration) designs. It was expected that on-chip interconnections would be the prominent bottleneck in terms of performance, energy and reliability. This occurs because the scaling of wires increases resistance, and consequently the wire delay and energy consumption, while tighter spacing affects signal integrity, and also reliability (XIE, 2010). The trend seen as the interconnection length is increased can be verified in Figure 2.12. The set of these parameters is crucial for the success of multi-core/SoC design. In order to attenuate the interconnection crisis, one promising option is the network-on-chip, where a general purpose on-chip interconnection network replaces the traditional design-specific global on-chip wiring, using switching fabrics or routers to connect IP cores or Processing Elements (XIE, 2010).

Figure 2.12 - Trend of total interconnection length on a chip.



Source: PASRICHA (2008)

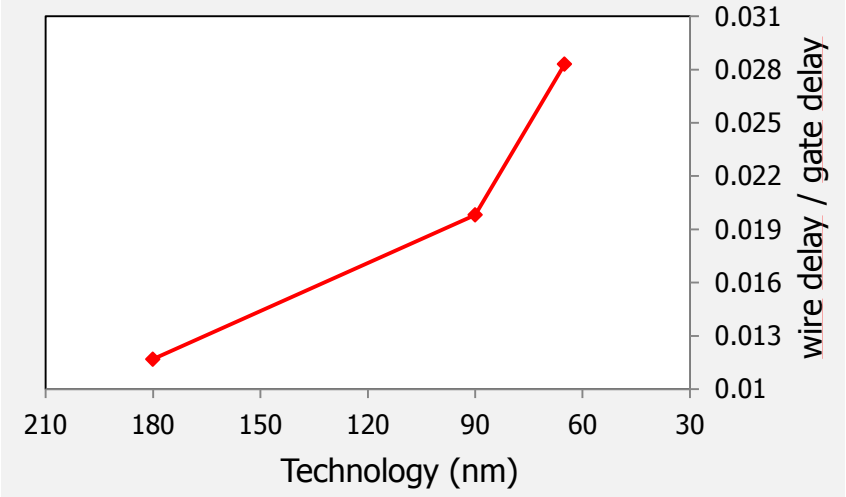
Technology scaling has allowed a large integration capacity. In such a context, a single chip can be composed of many processing elements to form the MPSoCs (SILVANO, 2011). Several heterogeneous elements can also integrate these systems, presenting different bandwidths and quality-of-service requirements (STENSGAARD, 2008). A fundamental issue in designing chip multiprocessor architectures with numerous homogeneous or heterogeneous functional blocks is the on-chip communication fabric (XIE, 2010), (BENINI, 2001), (KUMAR, 2002).

The industry has manufactured integrated circuits (ICs) composed of billions of transistors on a single die thanks to the fast development of new process technologies. With this evolution, the number of circuits on a chip, such as processors, memories, specific PEs and other devices has also increased, bringing an exponential growth in the complexity of their interactions (XU, 2010). Over the past years, NoC designs are studied as an appropriate solution for such complex hardware systems due to their scalability, parallelism and QoS (SILVANO, 2011), (YOON, 2010). Networks-on-Chip present several advantages when compared to other interconnection devices previously presented allowing, from a suitable design, the interconnection of hundreds of cores (GUERRIER, 2000), (DALLY, 2001), (AHMAD, 2005).

2.3.1 The impact of wires on the interconnections

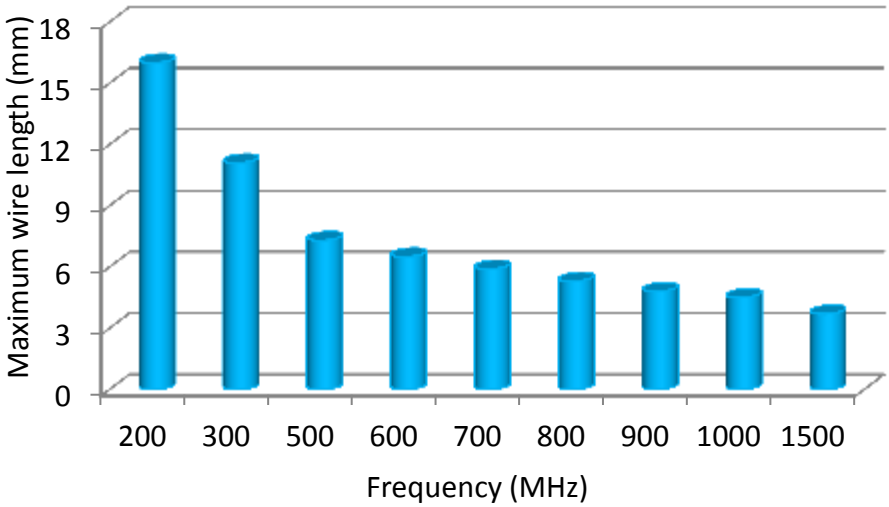
The shrinking of processing technology in the deep submicron domain aggravates the impact of the wires compared to the gates. This happens because while gate delays decrease, global wire delays increase due to increment of wire resistance (HO, 2001). With regard to the evolution of wires in the technology scale, Figure 2.13 presents the relation of wire delay per gate delay for different technological scales. As can be seen, the wires do not scale like the gates and indeed they prove to be the major problems in future designs. In this case, for recent technologies, the wire length pass became a major problem since the latency and power do not scale as the gates do. The parameters of this technology were obtained from (PTM, 2013) and to describe the electrical wire behavior, the RLC-π model (SAKURAI, 1983) for 65nm technology process was used. Using those same parameters, the maximum allowed frequency according to the wire length was verified. The results are illustrated in Figure 2.14. These results were also verified by the Hspice tool for 65nm process technology. For these experiments one repeater every two millimeters of wire length was specified. The repeaters are gates (e.g. two inverters) used to restore the signal, also called electrical buffers. As can be seen, systems running at a high frequency accept only short wires (MATOS, 2011). This means that only few cores can be crossed over by a wire without exceeding the clock cycle time. For example, if a SoC is composed of cores with an area around 1mm², for example an ARM1176 (ARM, 2013), only 4 cores can be crossed over by a wire for a frequency around 1GHz.

Figure 2.13 - Wire delay per gate delay according to the technology scale.



Source: elaborated by the author.

Figure 2.14 - Maximum wire length at every frequency.



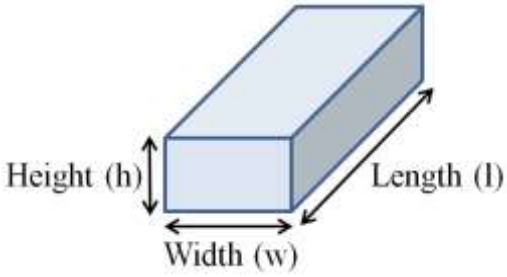
Source: elaborated by the author.

The wire delay can be calculated using a simple RC product (HO, 2001), (RABAEY, 2002). The increase in the wire resistance per micrometer occurs due to the increase of the resistivity and reduction in the wire dimensions when the technology shrinks, as can be observed in equation 1. As the chip area for the innovative MPSoCs does not reduce much, due to the need to add more hardware, the length of the global wires remains almost the same. With regard to the wire capacitance, it is directly proportional to the wire length (RABAEY, 2002) and this is inversely proportional to the distance between two wires (BOGATIN, 2003). As with technology evolution the wires are closer and the length does not reduce much, the delay is increased also due to the capacitance. These long wires also require the use of repeaters, which further consume power.

$$R/um = \frac{\rho}{wh} \quad (1)$$

where ρ is the resistivity, w is the width and h is the height, as presented in Figure 2.15.

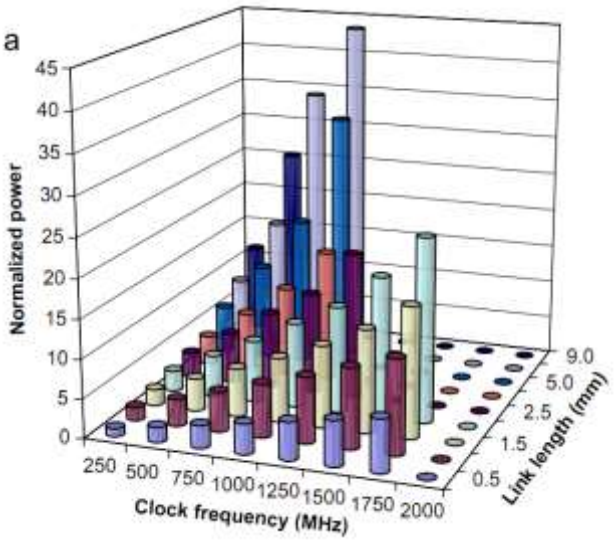
Figure 2.15 - Wire dimensions for an integrated circuit.



Source: elaborated by the author.

Finally, figure 2.16 illustrates the relation between wire length, and power and frequency. These results were obtained for 65nm process technology and 38-bit link. Values are normalized to the shortest link at the lowest frequency. In this graph, it is possible to see how the power consumption grows rapidly as the link length increases or the frequency is higher (ATIENZA, 2008). This occurs because when the links are pushed for high performance, many buffers are added to achieve the required results. If the frequency or wire length is pushed even further, the links are impracticable due to the timing violations or crosstalk concerns. From this analysis, the conclusion is that the impact of the wire power and delay depends on of the interconnection design.

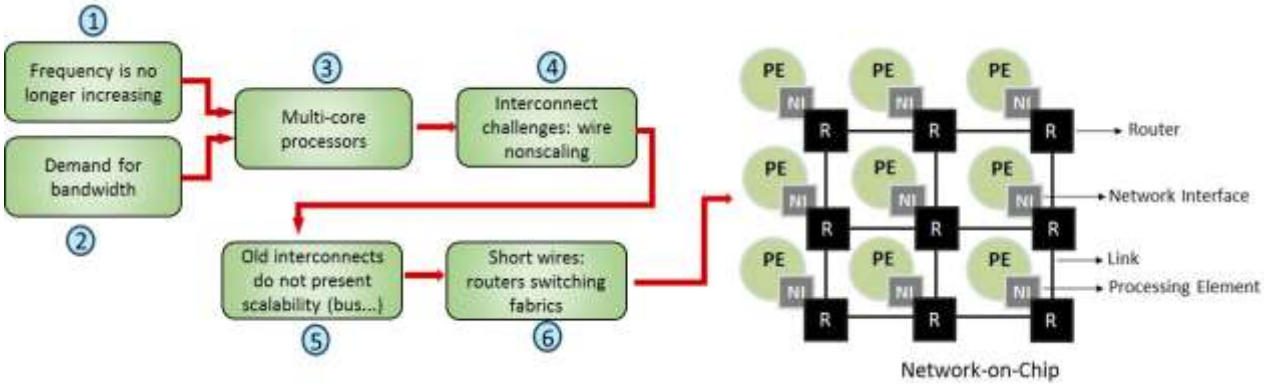
Figure 2.16 - Power consumption at different link lengths and operating frequencies.



Source: ATIENZA (2008).

As it has been shown, there are a lot of problems in the interconnections that need to be addressed. All circumstances previously discussed lead to new interconnection models. The networks-on-chip have emerged as a first solution for these paradigms. Many NoC solutions have been proposed in the literature and industry, confirming that old interconnection devices do not comply with the current design requirements. Figure 2.17 depicts a set of reasons that have led to the NoC structure.

Figure 2.17 - Reasons that point out to the network-on-chip.



Source: elaborated by the author.

Figure 2.17 presents an overview of the facts that point out the need for an interconnection device like an NoC. According to this figure, the first box refers to the problem related to the limitation of the Moore's Law (GOODWIN, 2007), once the maximum frequency and consequently, the performance of a single core, have not increased in accordance with the demands (Fig. 2.12). The request for more bandwidth leads to the multi-core systems. However, for this paradigm, old interconnection devices are not supported, due to the problems related to long wires, as already mentioned. The solution found was the NoC architecture, an interconnection device composed of routers where each router is linked to the others via short links. These observations will also be useful in the next chapter where the proposed solutions are presented. In the next section, the main features of the NoCs and the advantages and disadvantages of this solution will be commented.

2.4 Networks-on-Chip: interconnection model for complex systems

Firstly, this chapter will present the advantages and disadvantages of the use of NoCs and after, their main components and functions, such as topology, routing, switching, bufferization, arbitration and flow control.

As already seen, traditional interconnection solutions present physical limitations, compromising the scalability, performance, power consumption among others. A bus admits

only some PEs without a fall in the global performance occurring (ZEFERINO, 2002). The challenge is to establish a communication structure, able to meet requisites of performance and energy consumption for different applications (BENINI, 2001). In the networks-on-chip, the old problems do not occur because the interconnections between the ports are point-to-point and, in this way a large number of cores can be added in the network without affecting the system performance.

In summary, NoCs have a goal of obtaining alternatives for minimizing the problems related to other interconnection architectures, such as buses, dedicated wires, crossbars and others. Among the main advantages in the use of NoCs are:

- ✓ Parallelism (BJERREGAARD, 2006): due to the multiplicity of paths possible in an NoC;
- ✓ Structuring and connection managing in sub-micron technology (BENINI, 2002), (DALLY, 2001): utilization of short wires, point-to-point interconnections, reduction of the parasitic capacitances;
- ✓ Sharing of links (BOLOTIN, 2004), (DALLY, 2001): this allows increasing the link efficiency;
- ✓ Reliability (VELLANKI, 2004): due to the possibility of using many techniques to increase the reliability of the system;
- ✓ Scalability (GUERRIER, 2000), (BJERREGAARD, 2006): which allows adding more and more cores to the network without compromising the system performance;
- ✓ Reusability (BENINI, 2002): allowing the use of the same communication structure and existing cores in distinct applications;
- ✓ Routing decisions distributed (BENINI, 2002), (GUERRIER, 2000): allowing a greater counterbalancing in the use of network resources.

NoCs still have the possibility of instancing routers to compose different NoC sizes, scalable bandwidth and the use of dedicated BISTs (Built-In Self-Test). Only to illustrate, the buses present limited bandwidth and it is divided between all components, besides, they do not have good testability. However, as with any architecture proposal, NoCs also present some disadvantages, such as:

- ✓ Considerable silicon area and power consumption when compared to the other simpler solutions, because of the router and network interfaces required in these devices.
- ✓ Necessity to use network interfaces: the cores interconnected to each router need to have a mechanism to adapt the protocols among cores and networks;

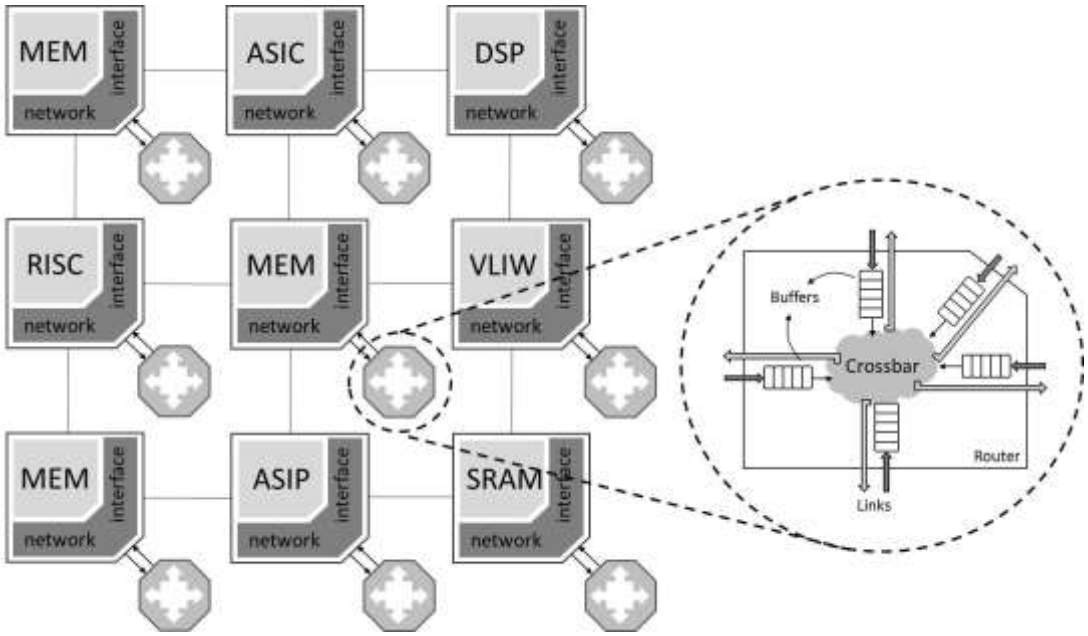
- ✓ Latency: due to the contention in the network or whenever the messages need to cross over many hops.
- ✓ Design complexity: many design parameters are to be defined that directly impact performance and energy consumption.

These disadvantages related to latency, power consumption and area are relative, since according to the number of cores and their bandwidth requirements, NoCs can obtain better results than other solutions.

NoC is a structure composed of a set of routers with links used to interconnect them. The routers are used to transfer the messages in order to reach the destination core. Usually, the message is divided in packets, and the packets in flits. Each router can have a core attached, be it a processor, memory, specific IPs or a cluster composed of processors and/or memories.

NoCs are classified according to the topology, arbitration, switching strategy, flow control and others. Figure 2.18 presents a 3x3 NoC and its main constituent parts. In this example, the buffering occurs only in the input channel.

Figure 2.18 - Example of a 3x3 NoC and the parts that constitute it.



Source: elaborated by the author.

The aspects of the NoC designs go from macro to micro architectures as depicted in Figure 2.19 (NICOPOULOS, 2010). In the macro-architecture, there are the features related to network, such as topology and routing. In the micro-architecture there are the aspects of the

router architecture, such as bufferization, arbitration, switching and flow control. In this work design features of both abstractions are considered.

Figure 2.19 - High-level overview of the NoC designs.

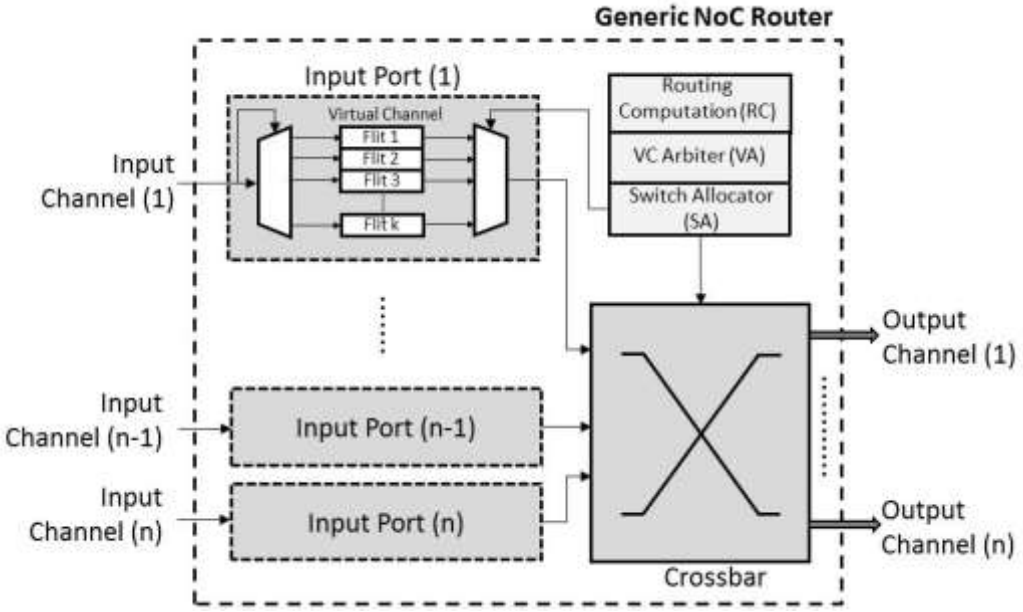
Source: elaborated by the author.

2.4.1 Router

A simpler router is composed of a switching device (crossbar), an arbiter, a routing control and ports which are made up of input and output channels. The channels present some memorization strategy to store the packets.

Figure 2.20 presents a modern router, and this model considers virtual channels in the buffering. In the router shown in Figure 2.20, the Routing Computation (RC) is responsible for directing the header flit of the packet to the corresponding output channel according to the routing algorithm. RC usually operates for each new packet, i.e., once defined the output channel for one packet, all flits are routed to the same output. The Virtual Channel Arbiter (VA) module arbitrates one packet/flit to be stored in the buffers among all the packets/flits requesting access to the same Virtual Channels (VCs) (NICOPOULOS, 2006).

Figure 2.20 - NoC router with virtual channel strategy.



Source: NICOPOULOS (2006).

In the architectures proposed in this work, virtual channels will not be used since this solution was proposed to improve the performance; however, it presents large power consumption. Other techniques for increasing the performance without consuming so much power will be presented. An interesting method of understanding the costs in the use of virtual channels was presented in (YOON, 2010). Their results are presented in Table 2.1. According to (YOON, 2010), the synthesis results were performed with Synopys Design Compiler with industrial standard-cell libraries for three different technology processes (90nm, 65nm and 45nm), and with target clocks set to 500MHz, 1GHz and 2GHz, respectively. The NoC power dissipations were made with Synopsis Primitime PX, assuming at each input port a uniform distribution of the data bits and a traffic load of 0.4 flits/clock-cycle for a flit-width of 128 bits.

Table 2.1 reports the power dissipation of each NoC router architecture, normalized to the reference architecture without virtual channels. According to the authors' conclusions, for short queues (buffer depth <= 8) the architecture with virtual channels causes substantial area overhead (59% - 136%) and power overhead (32% - 142%). However, they concluded that for large queues (buffer depth > 8), the impact in the use of VCs gets smaller as it increases. In these cases, it is interesting to use VCs. However, in this work only small queues have been adopted (buffer depth = 4 or 8) which do not justify the use of VCs.

Table 2.1- Power dissipation ratio for different technologies, values of VCs (2 and 4) and queues (4 and 8).

Buffer depth	4			8		
Tech(nm)	90	65	45	90	65	45
w/o VC	1			1		
VC = 2	2.03	2.42	1.92	1.68	1.32	1.36
VC = 4	N/A			2.51	2.33	2.37

2.4.2 Topology

The definition of NoC topology is determined by the manner in which their routers are interconnected. The topologies are classified as direct networks, indirect networks and irregular networks (PASRICHA, 2008) and the difference between them is in how the nodes, routers and links are connected to each other.

The topology also influences the routing of the network, since whenever a message arrives at a router it must specify the next path, whereby the message will be sent. The path of

the message is defined according to the neighborhood of each router, indicating the possible data flows in the NoC. The NoC topologies are classified as direct, indirect and irregular:

a) Direct Networks

In this class of topology, each node is attached to one router and this router is connected to neighboring routers through links. The nodes can be composed of PEs, memories or others. In this case, in order to attach more nodes in the NoC, the same number of routers needs to be added. In direct network topologies, the messages pass through several intermediate routers before reaching the destination node. Examples of these topologies are mesh, torus, octagon and hypercube.

Mesh is the most popular topology because it is regular, simple to implement and it allows the use of routing algorithms that do not require much hardware. In a 2D mesh topology, each router is connected to four neighboring routers (when this router is not at the edges) called north, south, east and west. Due to its regular grid-based architecture, the 2D-mesh topology has been widely used as the typical on-chip interconnection solution combined with tile-based architectures. Tile-based architectures deal effectively with parallelism through the replication of many identical blocks placed in each tile of a regular array fabric. Thus, the performance scalability is directly associated with the connection capability, rather than with the architectural complexity (GILABERT, 2009). The mesh topology can be reused, thereby allowing reduced design times. Moreover, mesh topologies have a high degree of performance predictability and allow good control of electrical parameters (GILABERT, 2009). The mesh topology however, may not provide an optimal solution for heterogeneous systems, as the results of this work will present. Some examples of proposed mesh architectures are SoCIN (ZEFERINO, 2004), Nostrum (MILLBERG, 2004) and SoCBUS (WIKLUND, 2003).

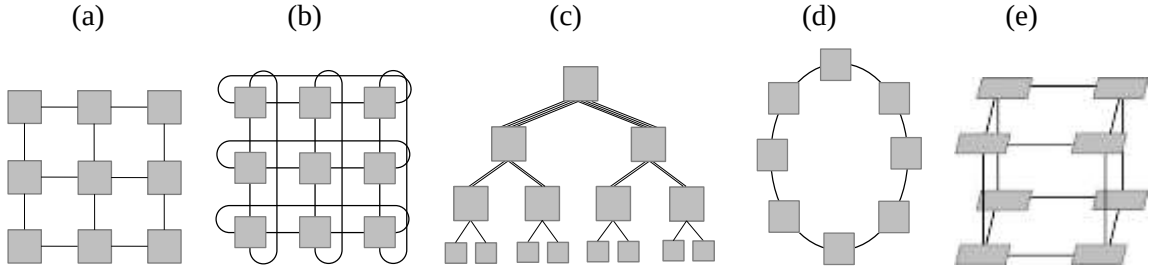
Torus topology is very similar to mesh topology but with the difference that the ports of the edge routers are not interconnected in the mesh architecture. The ports of the edge routers of the torus topology are interconnected to the opposite edge routers, as illustrated in Figure 2.21. The ring topology is the simpler topology, but this solution has its scalability limited once the performance decreases as more nodes are added to the network.

It is interesting to analyze the hypercube topology (Figure 2.21(d)). With the possibility of stacking the integrated circuit in 3D, this topology has been largely studied, and is now called of 3D mesh topology.

b) Indirect Networks

In indirect networks each node is interconnected to an external router, also interconnected to other routers. Examples of this topology are multi-stage topologies, trees and their variations fat trees and butterfly. In fat trees, the traffic is balanced because the links are increased the closer they are to the root, since it is where there is a higher concentration of traffic in the NoC (PASRICHA, 2008).

Figure 2.21- Examples of NoC topologies: (a) 2D mesh, (b) torus, (c) fat tree, (d) ring, (e) hypercube.



Source: elaborated by the author.

c) Irregular Networks

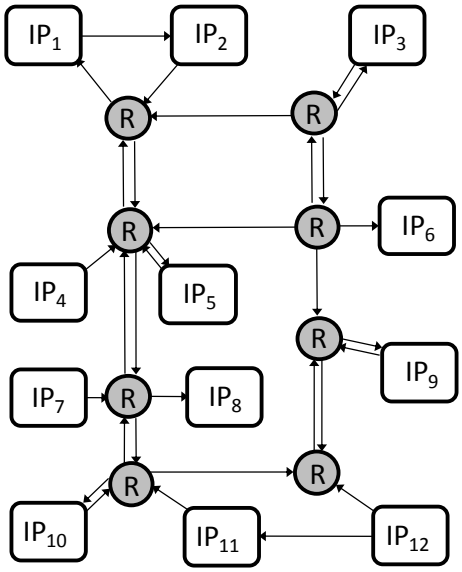
Irregular NoC presents an asymmetric topological structure (CHEN, 2012). Irregular topologies may be the combination of some regular topologies, as a mix of different type of topologies such as bus and direct topology, or direct and indirect topology in the same network (PASRICHA, 2008). Then, in such case, the final topology combines advantages of different solutions. Xpipes (BERTOZZI, 2004), SPIN (BENINI, 2002) and AEthereal (GOOSSENS, 2005) are also examples of irregular topologies. Some authors consider all hierarchical topologies as irregular ones (CHEN, 2012), (PASRICHA, 2008).

The proposed topology presented in this thesis concerns irregular network topology. Although the proposed topology is hierarchical, it is also classified as irregular since it can present variations in the number of cores in each cluster.

A customized topology is ideal since it is designed for a specific application and then obtains better results in terms of performance and energy than a regular topology (JERGER, 2009). The use of customized topology reduces the connectivity of a network justified by an application specific one, as in this case, a regular NoC with all interconnections would not increase the application performance besides presenting an increase in the area and power. However, the problem is when the application needs to be updated or that the core bandwidths can suffer alterations. Proposals of adaptive and irregular topologies can also be found, taking into account the need for an application (STENSGAARD, 2008), (CHAN, 2008). In this case, the efficiency and performance results are obtained for each application scenario. The

irregular topologies have a goal to reduce the distance between the nodes. In such cases, some routers are removed on the network to make a point-point connection between the processing elements with higher communication rates. In other situations, the routers can present an irregular number of ports. Figure 2.22 presents an example of irregular topology according to (STENSGAARD, 2008).

Figure 2.22 - Example of ideal irregular topology for an application specific.



Source: STENSGAARD (2008).

d) Hierarchical Networks

Hierarchical NoC topology refers to the networks with different communication levels (MANEVICH, 2013). The definition of hierarchical and irregular topologies is mixed in the literature. One manner in which it is easier to define the hierarchical topologies, is as homogeneous or heterogeneous. Homogeneous hierarchical topologies present the same structure on each level, i.e., all clusters are composed of the same number of cores as well as the number and distribution of routers in the topology, being identical among the levels. Examples of homogeneous hierarchical topologies are GigaNoC (PUTTMANN, 2007) and Multicross (DAS, 2009). GigaNoC has a NoC topology composed of a mesh on one level and wishbone bus on another level with four cores attached. In heterogeneous hierarchical NoCs the number of routers and cores is different on each level. Examples of this topology are HiNoC (HOLLSTEIN, 2006) and (CHOU, 2011).

2.4.3 Routing

The routing algorithms are associated with the NoC topology. The NoC routing will decide the paths on which the message will travel from the source to destination. There are some routing proposals whose purpose is to distribute the traffic in the network, avoiding the hotspots (congested channels), minimizing the NoC contention and improving the performance results (JERGER, 2009). The routing algorithms can be divided in many categories, such as, static, dynamic, distributed, centralized, deterministic or adaptive (PASRICHA, 2008).

Several routing algorithms have been proposed in recent years. However, the routings more commonly used are the dimension-ordered routings, due to the simplicity of the implementation. Dimension-ordered routing algorithms are also classified as deterministic, i.e., one message sent from node A to node B always travels along same path and are often used in the mesh and torus topologies. The most common example of deterministic routing is the XY (BOBDA, 2005). This algorithm firstly send packets in the X direction and only when it no longer needs to travel in that direction, it starts to travel in the Y direction until it arrives at the destination node. Once the packet has left the X direction, it cannot take this direction again. Deterministic algorithm, besides being simple, is frequently used in NoC architectures because it is deadlock free. Deadlock situation in NoCs occurs whenever there is a cyclic dependence between multiple messages. However, this guarantee causes a reduction in the number of possible paths along which messages can be sent.

Other routing possibilities are the adaptive algorithms. These algorithms have been largely studied in the literature (BOBDA, 2005), (MING, 2006), (HAIBO, 2007). The route decisions of these algorithms are taken in accordance with the traffic situation or are related to another condition, as for example the detection of faulty links in the network. In this manner, the adaptive routing algorithms can decide between one path and another with the purpose of reducing the latency and increasing the throughput in the network (MING, 2006), or to tolerate faults (HAIBO, 2007). However, in adaptive routing algorithms deadlock situations can be a problem. Some works propose solutions to outline these situations, and this is another topic extensively discussed in the literature (DUATO, 1993), (DUATO, 1995). Another challenge regarding the adaptive routing algorithms is to preserve the packet ordering in the reception, since each packet of a packet can travel by a different route to the destination (KIM, 2006).

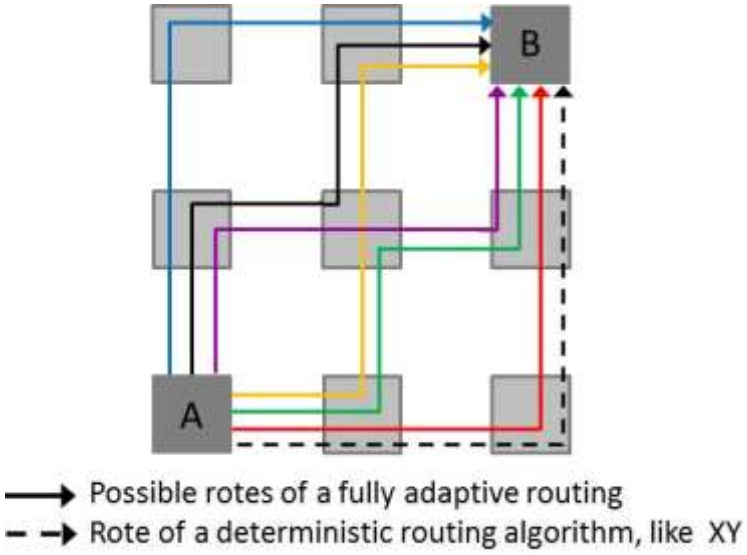
Other routing algorithms largely used are the table-based (BOLOTIN, 2007), (PALESI, 2006), (WANG, 2009), (MEJIA, 2009). These algorithms use a table inside each

router and from the table's information the next hop for the message is defined. Some adaptive routings use this technique to change the routing values according to the network availability (JERGER, 2009). Figure 2.23 presents the possibilities of minimal routes to send a message from A to B. According to these routes, the deterministic algorithm, like XY, presents only one possible path (the path shown by the dashed line), whereas for this case, there is a total of 6 possible minimal routes for the fully adaptive algorithms.

2.4.4 Switching

The switching is responsible for defining how the data flows in the network, i.e., how data received by an input channel is passed to one of the router's output channels. For a better understanding, it is needed to fully understand the granularity of the transferences of data in the network. A message is broken down in packets and each packet can be partitioned further into smaller units, called flits (flow control unit). Each flit is composed of one or more phits and each phit has the same width as the data physical channel.

Figure 2.23 - Examples of routing algorithms and their routes to send a message from node A to node B.



Source: elaborated by the author.

There are three main methods of switching:

- Circuit Switching (CS): in this strategy a path between the transmitter and receiver is reserved before the transmission of the messages. In this case, a direct physical channel between source and destination is established, which must be maintained until the conclusion of the communication. The reserved path is released only after the transmission of the message's trailer in the direction of the receiver (CHANG, 2006).

- Packet Switching (PS): In this method the packets are transmitted without the need to reserve the path by which the packet must travel. Whenever the links are free, the packets (or flits) stored in the buffers are conducted to the adjacent routers or to the destination node. There are three possibilities of PS, as follows:

    - Store-and-Forward (SAF): this strategy is used for short and frequent messages. In this case, the packets have a header with the information of the routing and each packet reserves the path until the destination. The entire packet is stored in the buffers during the path. A packet is sent from one router to the next only if the receiving router has buffer space for the entire packet. The buffer size in the router is at least equal to the size of a packet. Because of large buffer size requirements for this technique, it is not commonly used in NoCs (PASRICHA, 2008).

    - Virtual Cut-Through (VCT): this method presents the same concept of messages; however, in this case the packet does not need to be stored in its entirety. When the output channel is available, the remaining part of the packet is directly transmitted. This technique reduces the latency over SAF switching by forwarding the first flit of a packet as soon as space for the entire packet is available in the next router (differently than SAF that first waits for the entire packet to be received and then ensures that a sufficient number of buffer slots are available in the next router before initiating packet transfer). However, if no space is available in the receiving buffer, no flits are sent.

    - Wormhole (WH): this is the most common strategy and, in this case, the sending of messages is done by flits. In this method all flits follow the path reserved by the header and only after the sending of the trailer is the path released. The advantage of this method is that the buffers do not need to store the entire packet and, in this manner smaller buffer depth can be defined, especially since deep buffers insert considerable area and power consumption. With this strategy, a suitable buffer depth is defined in accordance with the required performance and power consumption. The disadvantage of this method is that once the path is defined for one packet,

other flits belonging to other packets need to wait for the end of the transmission of the current packet, unlikely in virtual channels, where the transmission of packets can be multiplexed.

- Virtual Circuit Switching (also called Virtual Channels - VC): This technique allows the reception of flits belonging to different packets through the same link. This way the path is not blocked by a single packet in the network. With this technique, it is possible to multiplex multiple virtual links on a single physical link. The number of virtual links that can be supported by a physical link depends on the buffer depth allocated to that link (PASRICHA, 2008).

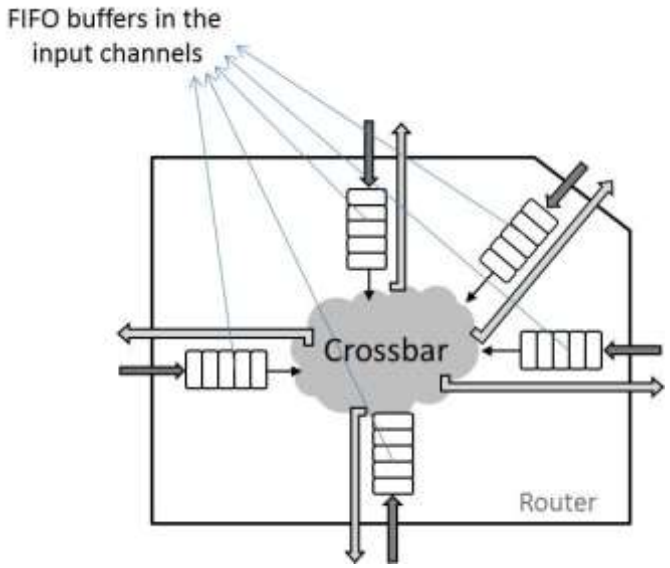
2.4.5 Bufferization

Bufferization in NoCs is required to store the message (packet or flit) when part of a path between the source and destination is blocked. In such case, the buffers ensure no losses occur, storing the message until the required output channel is free.

When a packet is sent by the network and needs to access a channel in use by other message, a bufferization mechanism is required so that the packet can be transmitted to the destination when the path is available. The storage of flits can be done either in input channels or output channels, or both. This strategy can be still centralized, independent or shared.

The most common bufferization is the implementation of independent buffers for each channel, as presented in Figure 2.24. A very common implementation is the FIFO buffer (First-In First-Out), since this alternative is simple to design. However, the use of a single channel with FIFO buffers can affect the performance of the network, as a single packet can access the channel while the message is transmitted. There are other strategies related to the buffers, such as SAFC, SAMQ and DAMQ buffers (TAMIR, 1992).

Figure 2.24 - Example of FIFOs in the input channels of a router.



Source: elaborated by the author.

In the centralized storage strategy, a single set of buffers can be partitioned among the channels of the router. In this case the buffer slots can be arbitrated for each channel, according to the receiving of the message requisitions in the router. This alternative can increase the NoC throughput, since those messages with high bandwidth are able to receive more of the centralized buffer slots.

Some works adopt bufferization solutions using adaptive techniques in order to increase the network performance or to reduce power consumption (NICOPOULOS, 2006), (AL FARUQUE, 2008), (MATOS, 2010), (MATOS, 2011). The adaptability in these cases is associated with the buffer depth of each channel according to the channel requirements in the router at a certain time. The buffers hold the main responsibility for the power consumption in routers (CHEN, 2003). Then, deeper buffers can ensure a higher communication performance. In this case, the channels with low traffic present smaller buffers, finding a balance between power consumption and performance.

2.4.6 Arbitration

The arbitration is responsible for selecting which input channel will receive a specific output channel. This mechanism is important when packets in different input channels have a dispute for the same output channel. The arbitration can be centralized or distributed. In the distributed mode, each output channel has an arbiter. There are several algorithms used by the arbitration, the simplest being the Round Robin which presents rotating priority. Other scheduling algorithms are also used, for example aging algorithms (FCFS, LRS), statistic priorities and others.

2.4.7 Flow Control

The flow control defines how the resources of the NoC are allocated to the packets that pass through the router. When multiple packets compete for the same resources of the network, a few techniques need to be considered. The most common flow control strategy is the use of buffers in the input or output channels to store the flits. In this case, a control based in a handshake defines when a flit can be stored in the buffers. If there is at least one free buffer slot, the flit can be stored. Two lines of control are used, one validation line (*val*) sent by the transmitter, indicating the intention to send a message to the receiver and another line of acknowledgement (*ack*) sent by the receiver, indicating the availability of buffer slot.

Comparison between the main interconnection devices

A comparison between bus, crossbar and a mesh NoC is presented in Table 2.2 on different aspects.

Table 2.2 - Bus, crossbar and mesh NoC architectures.

	Bus	Crossbar	Network-on-Chip (mesh)
Bandwidth	Degrades with the number of PEs.	Allows some concurrent communications	Allows several concurrent transactions without an impact on performance.
Throughput	Degrades with the number of PEs	Is only harmed when there is competition for the same output port.	Is only harmed if there is high contention in the network.
Latency	Small. After the beginning of a transmission, the time of sending is always the same.	Small. Degrades with the increase of the crossbar ports due to the long wires.	Presents extra latency. Is harmed whenever the # of hops increases or when the contention in the network is large. Depends on the mapping.
Spatial Complexity	O(N)	$O(N^2)$	O(N)
Power	Depends on the architecture. Is harmed with the # of PEs due to the parasitic capacitance.	As it does not present buffers, the power is smaller than the NoC up to a certain # of PEs.	Depends on the NoC architecture, but usually is high due to of the buffers and complex structure.
Scalability	Is harmed since the bus becomes slower with the increase in the number of cores.	Is harmed in function of the generation of long wires, large arbiters and MUXES when the crossbar size increases.	Is good due to the use of short links and the model based on packets (it allows pipeline).

	Bus	Crossbar	Network-on-Chip (mesh)
Clock	A global clock needs to be synchronized for the entire device.	A global clock needs to be synchronized for the entire device.	Allows high frequencies and the use of GALS.
Use of resources	The bus resources are used for all transactions.	Allows reusing the MUXEs and wires in the interconnections for the same output port.	Transactions by packets divide the router and link resources.
Arbitration	Central.	One for each output port.	Distributed.
Area	Smaller: does not use buffers in the control and presents simpler controls.	Smaller than NoC up to a limit. Does not use buffers but requires one arbiter for each output port.	Requires buffers for each channel, routing control, one arbiter and one crossbar per router, and other controls.
Reliability	Requires more complex techniques and extra hardware.	Requires more complex techniques and extra hardware.	As it presents redundancies in the use of resources, many techniques can be applied.

2.6 Considerations

In this chapter, the reasons for the introduction of networks-on-chip have been discussed. Some concepts of NoCs, their needs and functions in the current designs were also briefly discussed. After this, the main components of NoCs and some features considered in the proposed solutions were presented. Related to the NoC features, topologies, routing algorithms, switching strategies, bufferization, arbitration, and flow control were analyzed. In the next chapter, the first proposal of this work related to the topology will be presented. Different topologies were proposed in the literature and the advantages and disadvantages of each solution will be discussed.

3 2D HIERARCHICAL NETWORKS-ON-CHIP

One of the design challenges of current systems, concerns the communication costs of such applications. The new philosophy of systems-on-chip brought an enormous amount of integration issues and also new challenges in designing interconnection infrastructure. Despite this, the networks-on-chip, are one solution to the problems related to the SoCs, as mentioned in chapter 2.3 (wire non-scaling, more bandwidth, high frequency and many cores,) the regular NoCs may not perform as expected. This is due to the same problems as previously discussed; the systems even have a more complex composition, heterogeneous cores and demand even higher levels of performance. Therefore, other interconnecting solutions need to be proposed to interconnect processing elements instead of large and somewhat inefficient conventional networks-on-chips.

The performance and energy dissipation of a system is totally dependent on the network topology and how the cores are connected in the NoC. Selecting the network topology is one of the first steps in designing such types of complex interconnections, since the routing strategy, flow control methods and mapping cores of the network nodes, heavily depends on the topology (DALLY, 2003). The use of regular Network-on-Chip topology, such as mesh, torus and ring topologies, represents a prior decision, especially when neither are targeted to homogeneous systems nor are the system design characteristics (type and size of cores and traffic patterns) completely known at the beginning of the design phase. Besides this, in these regular NoCs routing wires tends to be a problem when used to interconnect large systems. Moreover, in systems with several cores, many hops can be required to send a single message to the destination, compromising the system latency. Higher performance and/or lower power dissipation are not possible from the general alternatives, regular topologies present poor performance and have a large power overhead (LAI, 2011), (MURALI, 2006). Also, a general purpose solution is not a good solution in the embedded system context where communication patterns are irregular and strongly application-dependent, while the cores are also completely heterogeneous in terms of size (CHAN, 2008),

(PALERMO, 2007), (YAN, 2008). These MPSoC features impact in the link's length and consequently in the network latency and in its maximum frequency.

Considering future systems with several cores, different small applications will be able to run in parallel, and only few interactions are needed among them. Moreover, for the same application, different bandwidths will be required for different regions of the system. In these heterogeneous scenarios, the communication among these regions usually has a lower communication rate. For this reason, a hierarchical topology is fundamental for the future of many-core systems in order to improve performance without using expensive interconnecting devices. As a result, it is possible to provide different levels of communications in a hierarchical manner.

A hierarchical NoC topology brings many advantages for complex MPSoCs since it can exploit the communication locality of the system, while still maintaining the NoC properties. The hierarchical NoC topology has also proved to not only reduce the number of hops when compared to a regular packet-switched topology, but also to provide a suitable bandwidth, power and QoS results.

In fact, hierarchy is not new, this strategy has been in use for a long time in different areas of computer architectures, as in the case of memory. However, the proposed hierarchy for NoCs of this work was not presented in the literature in the same way. The great advantage of the proposed topology is the large reduction in power when compared to other topologies due to the simple components used in the architecture and the performance gains obtained are a result of hierarchy.

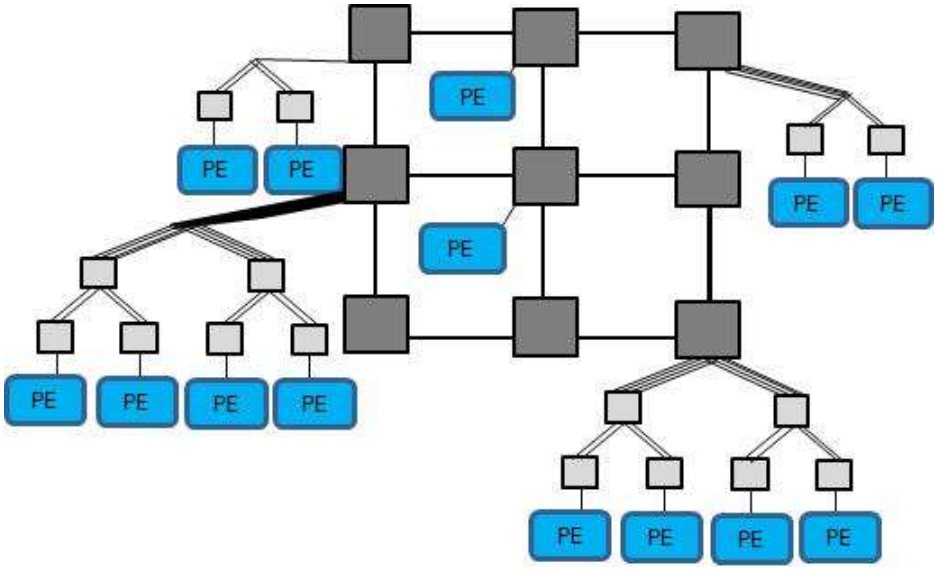
This chapter will discuss 2D topologies focusing on hierarchical NoCs. Firstly, in the next section, the state of the art 2D hierarchical NoCs will be discussed. Following this, the proposed topology, the architecture and an application specific tool for this topology will be presented.

3.1 State of the art 2D hierarchical NoCs

In the literature some hierarchical NoCs have been proposed and the main proposed topologies will be discussed in this section. HiNoC (HOLLSTEIN, 2006) proposes a network with two hierarchical levels, where the first level is a mesh and the second level is composed of fat-trees, as illustrated in Figure 3.1. The switching of the second level of that topology considers packets, and at the top level, it uses an asynchronous mesh structure. The asynchronous communication is handled by a handshake protocol. In the second level of this

hierarchy each fat tree is a general case of a binary tree, capable of providing a full connection with all of its leaf terminals. The fat trees attached to each mesh router may have different sizes and the communication at this level is completely synchronous. It uses an asynchronous algorithm on the global level and the packets are transmitted within this level by a virtual cut-through / wormhole switching scheme (HOLLSTEIN, 2006).

Figure 3.1 - An example of HiNoC topology.



Source: HOLLSTEIN (2006)

However, this solution needs to consider a particular routing algorithm. In that case, it uses a routing based in a combined static and dynamic metric. The routing algorithm is based on tables and the static routing tables are updated in order to adapt the routing for the new NoC configuration. Adjacent mesh routers exchange information about the current traffic in the network, sending control packets in free time slots. In that architecture, the cores attached to the fat trees are clocked synchronously, while in the mesh topology, the communications can be asynchronous.

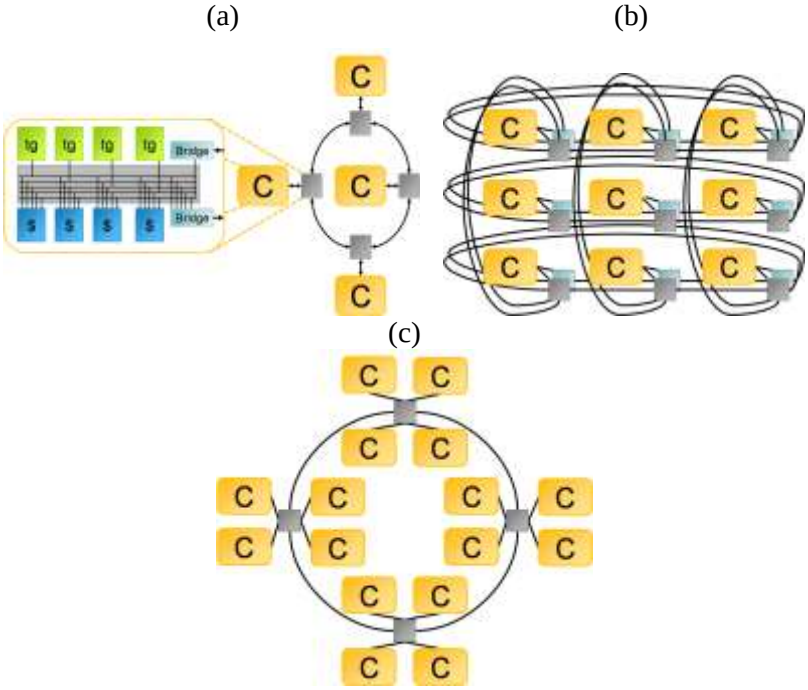
In the strategy presented in this work, the local level prioritizes high bandwidth communication, and for this reason circuit-switching is used on this level and GALS interfaces amongst the clusters can be used to adapt the different clock domains of each cluster.

In (GUERRE, 2010) eight different topologies were analyzed, three of them being hierarchical networks: CrossTorus, CrossRing and Multicross, as presented in Figure 3.2. In that comparison, it was observed that hierarchical NoCs show an area efficient solution. All hierarchical topologies analyzed in that work, consider a multibus architecture in each router

(see Figure 3.2(a)). These networks use a wormhole technique to transfer packets and do not consider virtual channels in the architectures. CrossRing (Figure 3.2 (a)) and CrossTorus (Figure 3.2(b)) present two levels of topology, the first topology of each network is a ring and a torus respectively and the second level is the multibus and bridges are required between the two levels. The MultiCross presents a particular configuration, since it is composed of 3 topology levels. In this topology, 4 clusters are connected around a ring, as can be seen in Figure 3.2(c). The first level of this topology, as the other hierarchical topologies, is a multibus. The second level is formed by a group of 4 clusters and the top level is a ring topology interconnecting these groups. In the results obtained from these topologies, it was observed that the mesh, which is the most popular topology, has a poor area efficiency compared to the hierarchical topology. Besides, MultiCross was the topology with the smallest area among the topologies studied, and MultiTorus was the topology that presented the best performance related to the hierarchical network. However, MultiTorus topology is very complicated to implement, presenting a limit in the scalability due to the wire routing related to the interconnections between the edge routers. According to the authors, MultiCross is a very good topology for multi core architectures. Nevertheless, that topology also presents scalability problems, since it considers a ring-based topology on the top level, which compromises the performance whenever the number of clusters and communications among them increases.

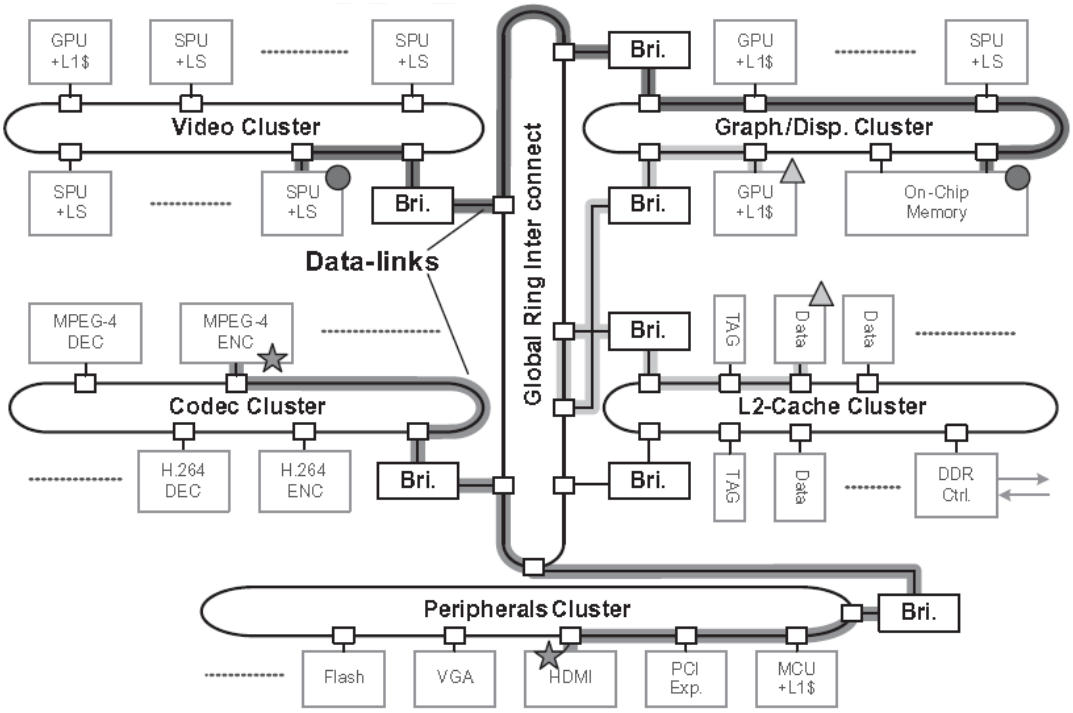
Other examples of hierarchical ring-based NoC topology are (CHOU, 2011) and (BOURDUAS, 2007). Chou proposed the HrNoC NoC topology. The hierarchy of HrNoC is composed of different levels of ring-based topology, shown in Figure 3.3. The authors said that only ring topology is used in the levels because that solution is specific for multicore video applications, in order to favor stream transactions. In this case, local rings facilitate burst and stream transactions. Bourduas presented a similar architecture ring-based topology, but in that case, the topology presented 3 levels, one composed of mesh routers and two other levels composed of rings, having a global ring interconnecting the local rings. However, as previously commented, both approaches present limitations, since only neighboring nodes are connected and, as a consequence, flits must travel through long paths, resulting in large hop counts and latencies.

Figure 3.2 - (a) CrossRing, (b) CrossTorus, (c) MultiCross topologies.



Source: GUERRE (2010).

Figure 3.3 - An example of HrNoC architecture.

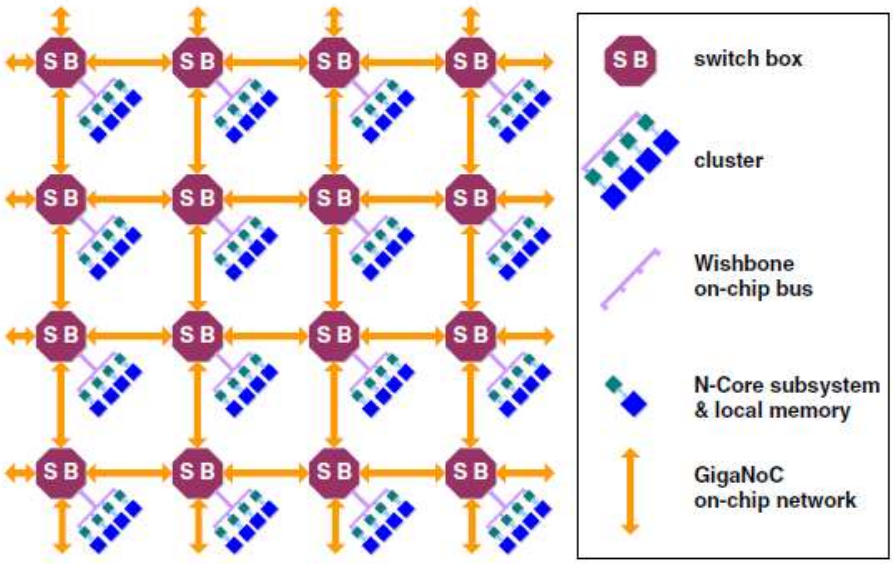


Source: CHOU (2011).

Another hierarchical topology analyzed by many authors is the combination of a mesh NoC topology with a cluster bus-based. GigaNoC (PUTTMANN, 2007) is an example of a

hierarchical NoC with cluster composed of buses. In those clusters, the cores are attached in a bus topology, shown in Figure 3.4. GigaNoC also uses transmission of packets by wormhole. The bus adopted in that topology is the Wishbone, which uses a multiplexed master-slave bus protocol with a round-robin arbiter that assures uniform access distribution. That architecture presents 4 types of flits: data flit, command flit, instruction flit and program flit. The transmission is started by a Command flit. This flit is responsible for specifying the total number of flits to be transmitted, since the Data flit increases the number of the flit sequence in every transmission, the receptor can identify the end of the transmission. An Instruction Flit is used to define some NoC features, for example, routing strategy. Program Flit is a special flit of data used for the initialization of local memory (PE level) during the boot loader phase. However, a topology based bus also presents some disadvantages, such as only one core being able to communicate at any one time. Besides this, the performance is impaired when the concentration of the PEs increases in each cluster. In the cluster architecture, it is more interesting to implement a very fast communication device among the cores, since the PEs that present high bandwidth may be allocated in the same cluster. GigaNoC also presents a high complexity when configuring the system due to the Wishbone bus protocol, which can increase the system's latency in function of its architecture and setup time.

Figure 3.4 - GigaNoC topology.

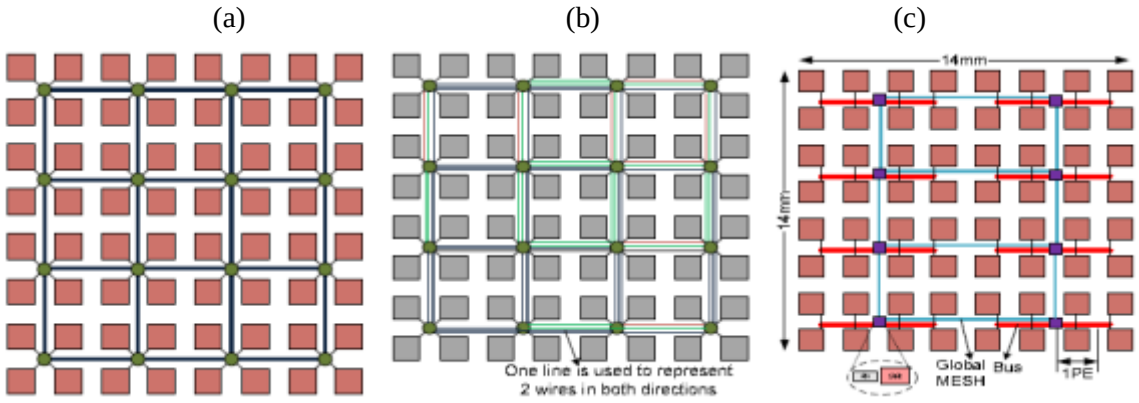


Source: PUTTMANN (2007).

Similar solutions of hierarchical bus-based NoC topology were also proposed by other authors, for example the solution presented by (DAS, 2009). In that paper, the tree topologies compared were: concentrated mesh, flattened butterfly and hybrid topology, as illustrated in Figure 3.5. In the concentrated mesh, each router is interconnected to 4 PEs. The hierarchical

network uses a high bandwidth shared bus for the first level of the local communication and a 2D mesh network for the global communication. Many proposed architectures consider a mesh topology at the top level because it is simpler to implement, presents scalability and a different routing algorithm can be adopted. Concentrated mesh reduces the number of routers and consequently the number of hops in the communications, thus yielding excellent latency savings over the mesh. The concentrated mesh is a good option for locality however; it has a low power efficiency. Flattened butterfly, presents problems with regard to the communication locality. Because of this, a hybrid topology bus base was analyzed. The hierarchical topology has a fixed number of nodes, equals 8, all of which share the same bus. This limitation will always occur when the bus topology is considered. However, even with this limitation, among the verified topologies in that work, the hierarchical topology obtained the best results in energy delay product considering synthetic traffic.

Figure 3.5 - (a) Concentrated mesh, (b) flattened butterfly and (c) hybrid topology.



Source: DAS (2009).

In the comparison of these topologies, the hierarchical solution with the shared bus was that which presented the best results in power consumption. However, the hierarchical topology presented lower throughput than the radix topologies. In order to improve the throughput, the authors proposed a priority channel among the virtual channels to transfer multiple small flits. This special channel allows, for example, small packets composed of only a few flits of control, therefore using the channels more efficiently.

The hierarchical topology, which will be presented in this work, assumes that the highest communication volumes need to use a communication device with larger bandwidth than the higher hierarchical levels. In such case, the proposals that consider cluster bus-based are not a good idea, as already discussed. Another problem found in some proposals of the literature is the use of topologies with scalability limitations, as is the case for ring topologies.

In the solutions reported in this section, only those solutions for ASIC (Application-Specific Integrated Circuit) are reported, i.e., proposals for FPGA (Field-Programmable Gate Array) implementation were not discussed. FPGA designs present different strategies, since they integrate programmable interconnects and logic components. Besides of this, as these different design strategies present specific goals, a comparison between them would not be appropriate. As the proposals of this work are designed for embedded systems composed of several cores, an ASIC design is the most suitable, since still presents some advantages when compared to FPGAs, such as lower power consumption and high operating frequency (VALDERRAMA, 2011). Table 3.1 presents the resume of the proposals presented in this section and their main benefits and limitations.

Table 3.1 - Resume of the hierarchical proposals found in the literature.

Author	Strategy	Benefits	Limitations
(HOLLSTEIN, 2006)	Global interconnection: mesh routers Local interconnection: fat-tree	✓ Qos (using dedicated virtual communication channels) ✓ Debugging and Testing	o Complex and expensive architecture in terms of implementation (tables for routing) o Power o Area
(GUERRE, 2010)	Global interconnection: ring NoC Local interconnection: multibus	✓ Area	o Scalability problem on the global and local levels o Performance
(BOURDUAS, 2007)	Global and intermediate levels: ring NoC Local interconnection: mesh	✓ Area ✓ Power	o Scalability problem on the local level
(CHOU, 2011)	Global interconnection: ring Local interconnection: ring	✓ Favor stream transactions ✓ Area ✓ Power	o Scalability problem on the global and local levels
(PUTTMANN, 2007)	Global interconnection: mesh routers Local interconnection: bus	✓ Area ✓ Power	o Configuration complexity o Setup time o Power Consumption
(DAS, 2009)	Global interconnection: mesh routers Local interconnection: bus	✓ Energy	o Performance
Proposed Solution – HiCIT	Global interconnection: mesh routers Local interconnection: crossbar	✓ Performance ✓ Power ✓ Energy ✓ Area	o Scalability problem on the local level but on a much smaller scale than the other solutions.

In the next section, the HiCIT topology and the advantages of this proposal compared to the other solutions of state of the art previously presented will be introduced.

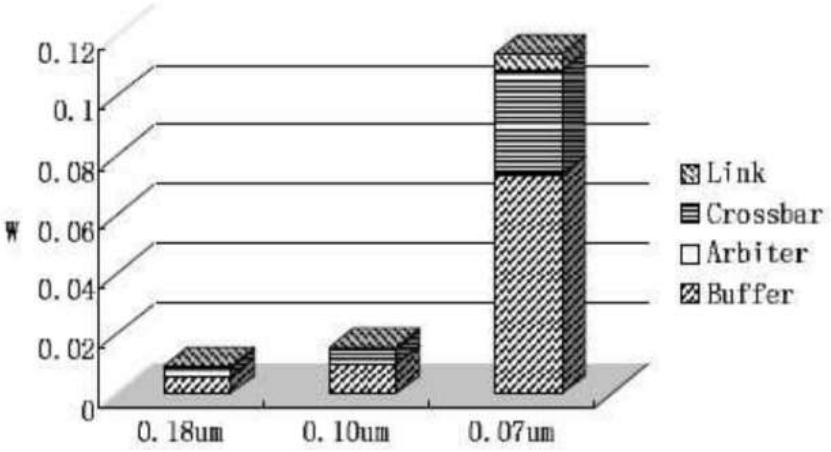
3.2 HiCIT topology

As discussed in the section above, some researchers observed the need to obtain a hierarchical network-on-chip to meet the requirements of current and future MPSoCs. However, as shown in Table 3.1, it is possible to see that some authors proposed an efficient solution in some aspects, while leaving other aspects with a lot to be desired. No authors were able to show efficient solutions in all aspects. Some solutions improve the performance but pay a high price, consuming too much power. Others present simpler techniques allowing reduction in the area and power consumption, but do not obtain relevant gains in performance. In this manner, it is clear to see that a good alternative to interconnecting complex systems is to provide different levels of communications in a hierarchical manner.

The main advantage of a hierarchical architecture is the exploration of the communication locality of the applications. So, the purpose is to map the cores that require high communication rates in the same cluster and the clusters are then interconnected on a communication global level. One possible way to obtain higher performance in communications would be to use a circuit-switching solution, employing crossbar switches on an interconnection level. In order to circumvent the scalability problems of the crossbar component, the usage of a hierarchical topology using crossbar switches for local communication and packet-switched routers for global communication, seems to be a viable solution.

Some researchers have presented some restrictions to using larger crossbars due to their quadratic cost increase. In (DAS, 2009), the authors demonstrated that crossbars present the largest power consumption in a router, however, these considerations were made for a crossbar with channel width equals 512 bits, and do not mention the buffer depth considered in the channels. Many other authors showed that the buffers hold the main responsibility for the power consumption in a router (YE, 2002), (CHEN, 2003), (CHANG, 2006). Figure 3.6 illustrates the power consumption of a conventional NoC architecture. In this example, VCs were not taken into consideration, which would represent an even larger power consumption of the buffers. According to this figure, the buffers were the power consumers with the largest leakage, dissipating approximately 64% of whole power budget. With regard to dynamic power, the buffers consumption is also high, and it increases rapidly as packet flow throughput increases (YE, 2002).

Figure 3.6 - Power consumption of each part of the router.



Source: CHANG (2006).

In order to understand the crossbar scalability, some experiments were analyzed beforehand to define the proposed NoC topology. Firstly, the possibility of using crossbars to compose the hierarchical NoC due to the possible advantages in area and power compared the current complex routers was verified. Together with the analyses presented in Chapter 2 related to crossbars, other evaluations in terms of power consumption and area for 65 nanometers process technology were measured. Figure 3.7 presents the area and power consumption results when the number of attached cores is increased. For these analyses, it was compared the crossbar with a mesh router with buffer depth equals 4 and 8. The router mesh architecture considered is the SoCIN architecture (ZEFERINO, 2003). For each core, one router was required in the mesh topology while in the crossbar, one port is increased with the number of cores. For example, a 4x4 crossbar was compared to 2x2 mesh NoC, since both allow 4 interconnecting cores. Links of both crossbar and mesh were set to 16-bit link wide. In order to obtain the impact of the wires in the design, cores with area of 0.488mm² were inserted (similar to the area of an ARM946E-S). These results were obtained with the RTL Compiler and First Encounter tools for 65nm process technology and operating frequency of 500MHz (this frequency was chosen since large crossbar does not operate with larger operating frequencies).

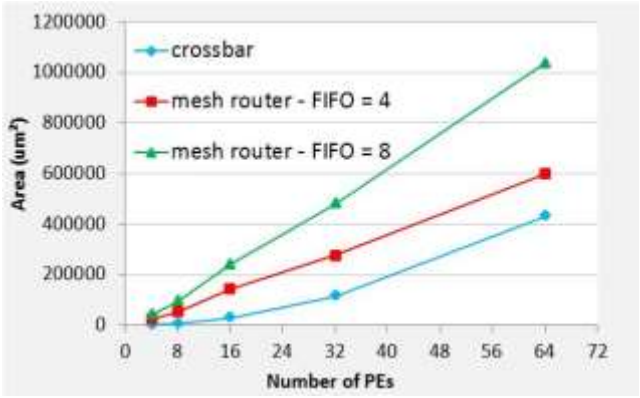
These experiments revealed that a crossbar switch with a few tens of ports can provide a lower power consumption than a mesh NoC (Figure 3.7(b)). For small buffer depths (FIFO=4 slots, for example) the crossbar has a smaller power consumption for few cores (up to around 30 cores), however, considering larger buffer depths in the ports, the crossbar architecture presents advantages for systems composed of several cores. In the area

comparison, crossbar is much smaller than a mesh topology even when the routers present larger buffer depth.

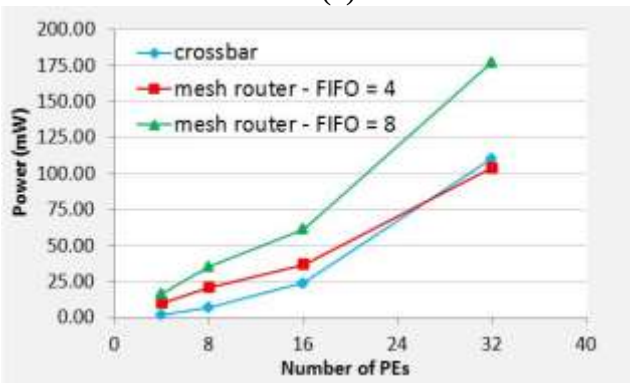
The results presented in Figure 3.7(c) were investigated in order to understand, beside the wires (that will be presented), what component of a crossbar is responsible to increase the power consumption of the component. As can be seen in these results, the switching consumes more power than the arbiter (comparing with the arbiter implemented in the HiCIT architecture).

Figure 3.7- (a) Area and (b) power consumption analyses for crossbars. (c) Power consumption for the parts of a crossbar.

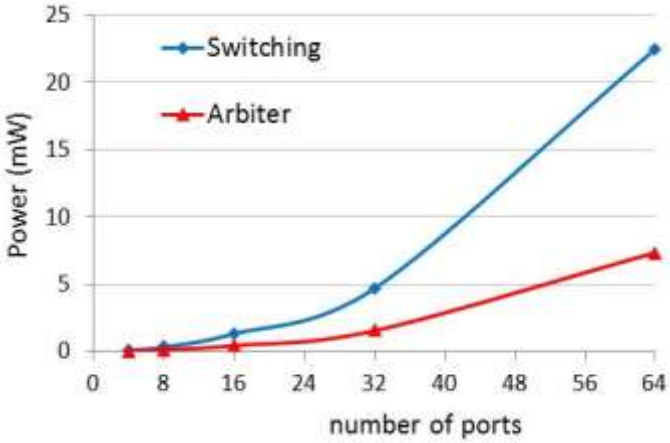
(a)



(b)



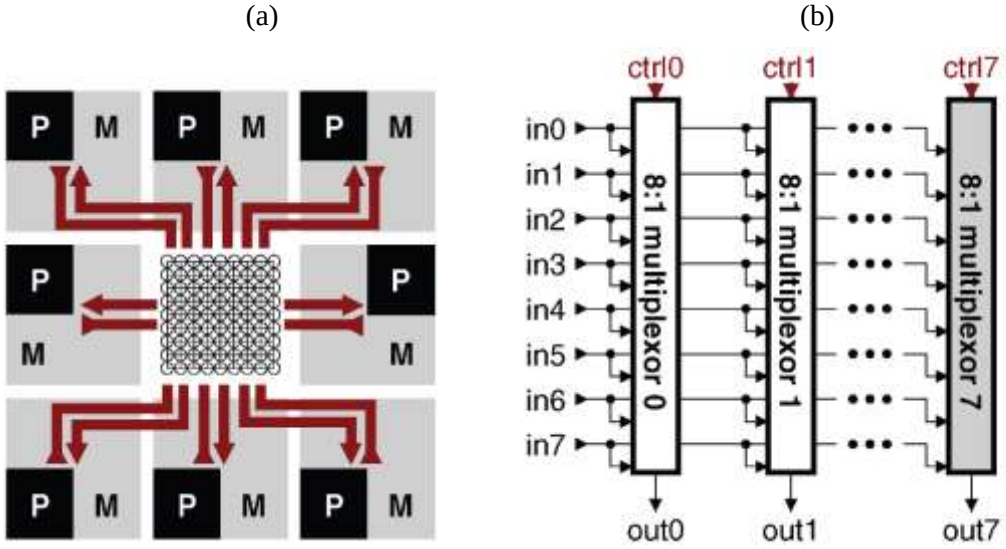
(c)



Source: elaborated by the author.

A recent study considering the number of nodes interconnected by a crossbar was presented in (PASSAS, 2010) and in (PASSAS, 2012). In these analyses the authors prove that, contrary to common belief, it is possible to obtain an efficient crossbar design even to tens of ports, hence substituting many complex routers. In this experiment, the authors have implemented a crossbar with a centralized floorplan, where the crossbar is placed in the center of the tiles (Figure 3.8 (a)). According to the authors, a centralized crossbar was preferred because it is easier to control and the total lengths of global links are smaller than the distributed crossbar. A 1-bit radix crossbar connects N 1-bit input ports with N 1-bit output ports using N multiplexers N:1 each. Each replica of switches is a bit slice (Figure 3.8(b)).

Figure 3.8 - (a) Matrix using centralized crossbar where *Ps* denote processors and *Ms* memories. (b) bit slices (1-bit radix-8 crossbar).



Source: PASSAS (2012).

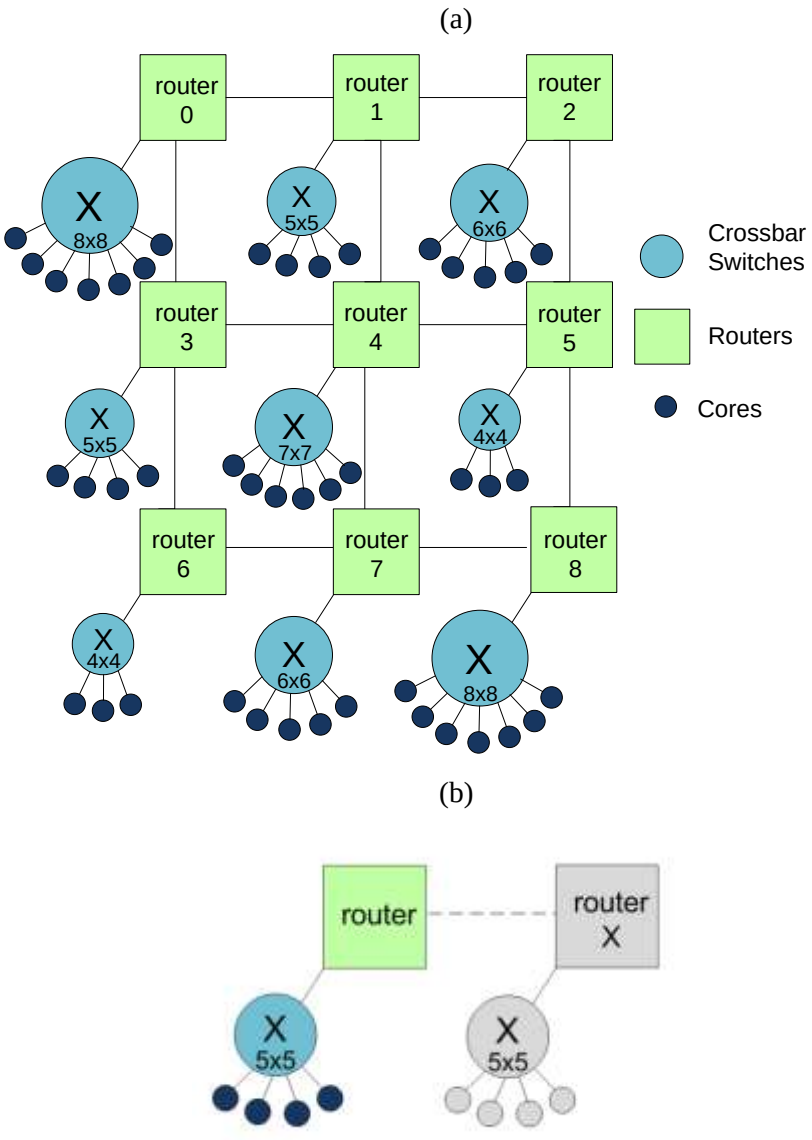
In that crossbar design the multiplexer gates were connected in a binary tree instead of a chain since the delay is lower. This proposal was implemented with a three-stage pipeline, which allows integrating more than 100 nodes. The results were obtained for 32 bit link wide, 90nm CMOS standard-cell and operating frequency equals 750MHz. From these parameters, the total power consumption grows as $O(N^2)$ for crossbar with N below 32, while for larger N, it grows with super-quadratically with N, due to the large overhead of input lines internally to the bit slices. In comparison with a torus NoC with four VCs per port and each VC with two-flits deep buffers, the crossbar reduces the latency of global communication by removing excessive delays due to intermediate hops, while it increases the latency of local communication, since the latency between neighboring tiles in a mesh is only three cycles compared to 11 cycles for this crossbar design.

Considering the proposed state of the art hierarchical architectures for on-chip interconnections, the primary justifications for not using a crossbar as an alternative interconnection for reasons of power consumption have lost weight. It is clear that there is a limit in the size of a crossbar that can be explored, and this is justified in a clustered communication approach. As one can verify, from of these different analyses, the limit in the crossbar size depends on many parameters, such as architecture, size of cores, operating frequency and others. In this work, from results obtained in Figures 2.10, Figure 2.11 and Figure 3.7, one considers small crossbars those with N smaller than 16.

HiCIT architecture is a soft-core described in VHDL (VHSIC Hardware Description Language). In the next subsections, the components that compose the HiCIT (Hierarchical Crossbar-based Interconnection Topology) architecture will be presented. In this architecture small clusters are used that provide point-to-point connections and allow a reduction of the area and power consumption yet improving the communication performance of the system. HiCIT explores the communication locality of the system and reuses wires wherever possible, and due to these features, this proposal obtains gains in different aspects compared to a conventional mesh network (MATOS, 2011a), (MATOS, 2012). The crossbar switches are simplified, and provide fast communication between the cores. HiCIT is based on this context where, on the global level, only communications with low bandwidth are required. In this case, each application needs to have an appropriate mapping for HiCIT and adequate crossbar granularity (MATOS, 2011a).

Figure 3.8 shows an example of an application using the HiCIT topology. In this example, 55 cores are connected in a network-on-chip, with nine clusters composed of routers and crossbar switches of varying sizes. This can easily be confused with a concentrated mesh proposal, but as illustrated in Figure 3.9(a), HiCIT is largely different to a router with many ports with each one connected to one PE. A concentrated mesh uses one port with many buffer slots for each PE attached to the switch. The gains of HiCIT are obtained mainly due to the non-use of buffers on the local level. The scalability of this proposal is achieved not by the increase of the ports of a crossbar switch, but rather by replicating the crossbar structure with a limited granularity, as illustrated in Figure 3.9(b).

Figure 3.9 - (a) Example of HiCIT architecture. (b) Scalability of the proposed topology is reached increasing the set of router + crossbar.

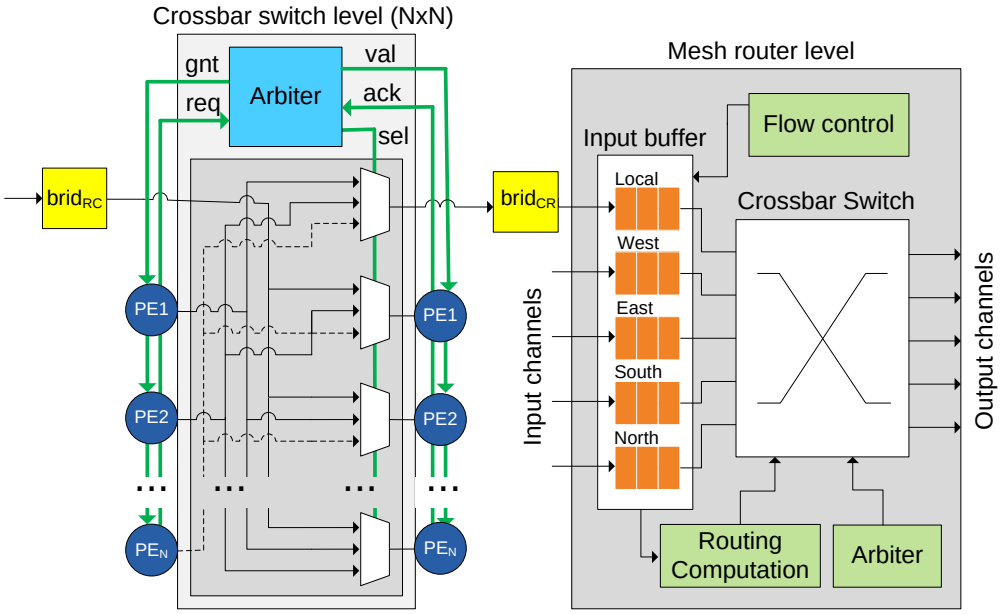


Source: elaborated by the author.

3.2.1 Local level architecture

The use of the proposed crossbar switch instead of the conventional NoC routers brings many advantages, since the PEs are directly connected to each other. In such cases, one does not need to use buffers and extra controls like in a conventional NoC router. As the local level does not have buffers, the hierarchical interconnection proposed here can reduce the power dissipation of the NoC. In addition to the proposed topology, the performance of the system is improved, since the cores are in the same cluster, the latency to transfer the packets for many routers is removed, and a simple switch protocol is used in this case. Figure 3.10 outlines the HiCIT architecture, presenting the two levels of this topology.

Figure 3.10 - HiCIT architecture composed of a crossbar switch on the local level and a mesh router in the global level.



Source: elaborated by the author.

The crossbar switch (Xbar) architecture aims to allow the communication between the cores through multiplexer switches. Parallel communications can occur if the data is sent for different output ports. However, if there is a conflict for the same port, a Round Robin (RR) arbiter is used in this architecture, to avoid starvation. A handshake flow control is applied for each port, using a requisition signal and a granted signal.

Each crossbar switch is composed of a set of multiplexers and one arbiter for each output port. Therefore, when a core belonging to a cluster needs to communicate with another internal core, it sends a request to the arbiter with the ID (identification number) of the required destination core. The data is only sent to the destination core if it is available to receive it.

The implementation of the RR algorithm takes only two cycles to verify the requisition of each port, independent of the crossbar switch size. In the first cycle, the arbiter verifies all requisitions of the input ports for an output port, and in the second cycle, it meets the next requisition, following in order. As only small crossbars are used in this topology ($N < 16$ for average core area $< 1mm^2$), the maximum operating frequency is guaranteed. The crossbar switch design is totally configurable, allowing one to define the number of ports of the crossbar and the data link width.

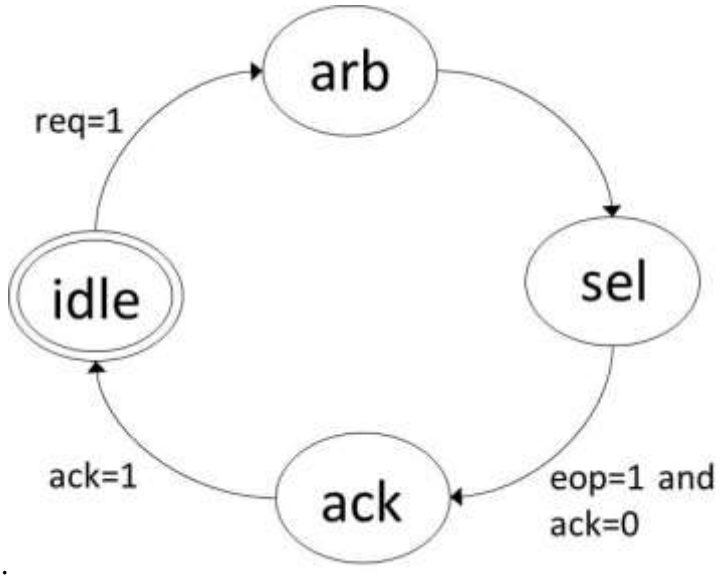
The crossbar communication uses a sending requisition, followed by a transmission acknowledgment sent by the arbiter of the required output port. Once this acknowledgment has been completed, a circuit is closed between transmitter and receptor. In short, this

architecture is composed of a set of wires, multiplexers and control logics. These components have the arbitration and protocol control of each output port as function.

Each output port receives a *req* signal and the requisitions of all input ports are concatenated in a bus of *n* elements (*n* is the granularity of the crossbar). From the *req* signals, the arbiter set the *gnt* as soon as it is ready to receive the message. In the sequel, the control logic defines the multiplexer selection (*sel* signal), closing a direct path between the chosen input port and the output port.

The state machine that controls the arbitration and data flow control of each output port is shown in Figure 3.11. Initially, one requisition waits in the idle state. Whenever one *req* is equal to 1, a transition for the *arb* state occurs and a priority table is verified in order to define the request to be served. This query takes 1 clock cycle. After this, it passes to the *sel* state, responsible for modifying the multiplexers according to the select signal. A valid signal (*val*) is sent to the destination port indicating the intention to send data and an acknowledgment signal (*ack*) is retransmitted to the source. The data is sent in the same model of packets, since it facilitates when the data needs to be transmitted to the mesh NoC. When the end-of-packet (*eop*) is sent, control passes to the next state (*ack* state) and it waits until the reception of the ack signal that indicates the consumption of last flit. While the ack is not received, it remains in this state. Once the ack signal is sent, it passes to the *idle* state again.

Figure 3.11 – State machine to control the data transfer of the crossbar switch.



Source: elaborated by the author.

3.2.2 Global level architecture

On the global level, mesh topology was used for the same reasons as previously commented, as this topology is scalable and easy to implement. The NoC router is based on the RASoC (Router Architecture for SoC) architecture (ZEFERINO, 2004) which composes the 2D mesh NoC called SoCIN. The routers have five bi-directional ports, four to connect to the other routers (North, South, West and East) and one to the cluster, in this case, the hierarchical topology. Each port has two unidirectional channels and each router is connected to four neighboring routers. This architecture is parameterized in three dimensions: communication channel width, input buffer depth and routing information width. The architecture uses the wormhole approach and a deterministic source-based routing algorithm (XY). The XY routing algorithm is capable of supporting deadlock-free data transmission, and the flow control is based on the handshake protocol. The wormhole strategy breaks each packet into multiple flow control units called flits, and they are sized as an integral multiple of the channel width. The first flit is a header with the destination address followed by a set of payload flits and a tail flit. In order to indicate the flit information (header, payload, and tail flits) two bits of each flit are used. There is a round-robin arbiter at each output channel. The buffering is presented at the input channel only. Each flit is stored in a FIFO buffer unit. The input channel is instantiated to all NoC channels and thus, all channels have the same buffer depth defined at design time.

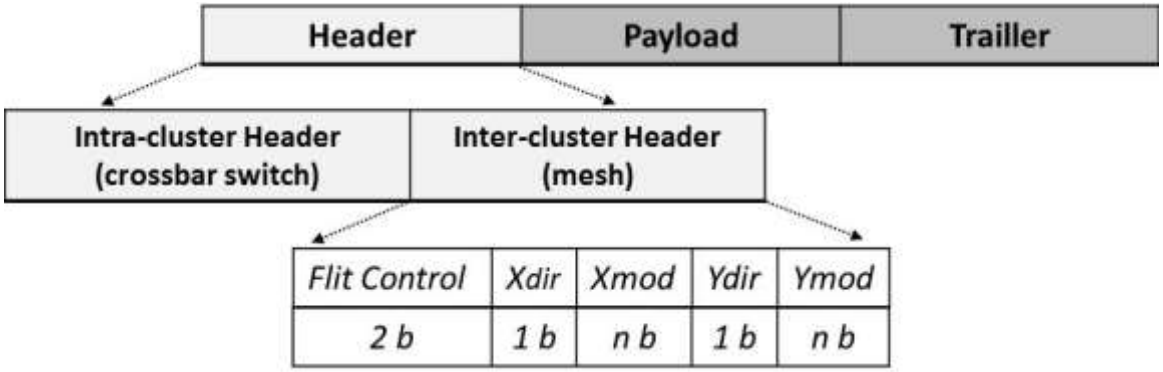
This router presents very simple architecture since it does not use virtual channels and the routing is the XY, which does not require internal tables within the router. Once a hierarchical topology has been defined, the router architecture can be simpler since the performance gains will be achieved as a result of the communication locality and the maximum bandwidth of each level. In this manner, the use of a low-cost architecture allows a reduction of the area and power consumption, when compared with many other complex and expensive NoCs (JERGER, 2008), (MODARRESSI, 2009).

3.2.3 Bridge architecture

Bridges are used in the connection between the global and local levels, leaving this boundary transparent at each hierarchical level (MATOS, 2012), (MATOS, 2013a). HiCIT has two different packets protocols: the local protocol and the global protocol. The global protocol uses a typical mesh topology header. In this case, the header flit contains the information of the XY routing algorithm and of the local header (with the identification of the destination core).

The messages in the HiCIT topology have a packet format even on the local level. This was defined to simplify the transmission on both intra and inter levels. The packet format also has the purpose of limiting the transmission time of a determined node. In the inter cluster communication, the header needs firstly, to identify the destination cluster, and then the node of the cluster. In this case, the header is composed of a local (intra) and a global (inter) header, as presented in Figure 3.12.

Figure 3.12 – Packet format of the HiCIT architecture.

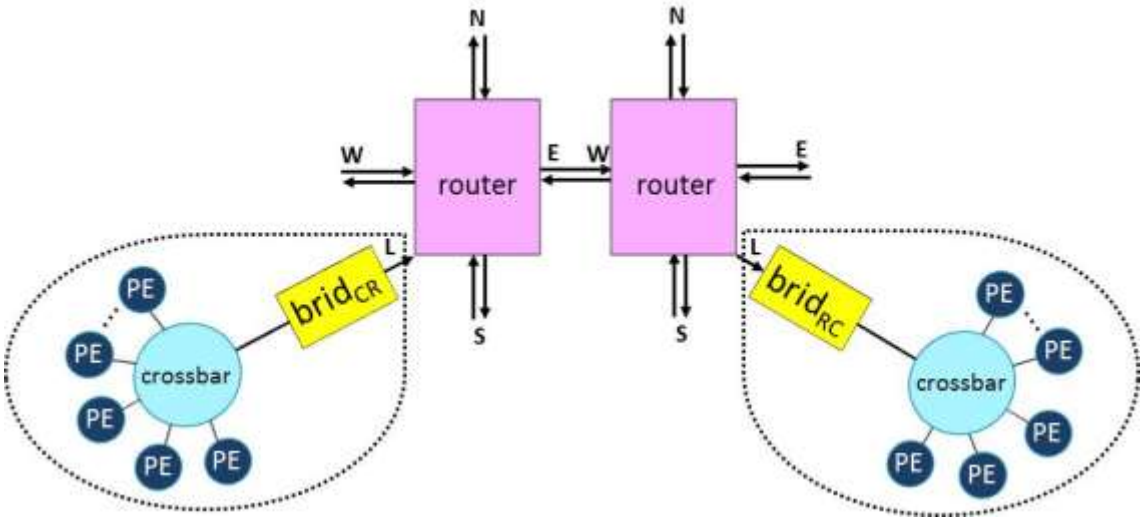


Source: elaborated by the author.

The header of the inter-cluster contains the control that defines the begin-of-packet (bop), end-of-packet (eop) and the routing information. The routing information presents the direction of each coordinate (Xdir and Ydir) and the module which specifies the number of hops to travel in the X and Y coordinates (Xmod and Ymod, respectively). One bit is used to define the direction of each coordinate and n bits, the number of hops. This number of bits depends on the network size and maximum distance between source and destination.

First and foremost, the network interface divides the data in flits. The first flit contains the header and the last flit presents the Flit Control, indicating it as the trailer, (see Figure 3.12). Hereafter, the requisition signal (*req*) is set, so, the crossbar switch identifies the destination node, and the arbiter checks the possibility of transmission, closing the communication if the crossbar output port is available. The output port can be interconnected to another node or to the cluster router (when the communications are inter-clusters, changing the communication level). In the crossbar, once the communication is closed, the packet is transmitted as a burst. Figure 3.13 presents the interconnection of the bridges ($brid_{CR}$ and $brid_{RC}$) between the levels.

Figure 3.13 - Two levels of HiCIT architecture and the bridges used to interconnect them.



Source: elaborated by the author.

The bridges have the function of adapting the protocol used at each level. In this case, a bridge from the crossbar switch to router direction (bridCR) is used to remove the controls of the local protocol. When a message arrives at the destination cluster, a bridge from a router to crossbar switch direction (bridRC) is used to remove the controls and header parts of the packet are used on the top level. The bits referring to the cluster header are transferred for the correct control of the cluster. The bridge bridCR adapts the handshake protocol to allow the sending of data by the network. In this manner, the bridges have a similar function as the network interfaces. Moreover, the use of GALS (Globally Asynchronous Locally Synchronous) interfaces in these systems is extremely important. Furthermore, as the clusters are heterogeneous, each one will probably have a different maximum operating frequency. Using GALS, a synchronous clock is defined for each cluster while the communication between them is achieved asynchronously.

In order to allow the cores in the same system to be able to operate with GALS strategy, synchronization interfaces are required. GALS interfaces are proposed to solve problems related to distribution of a skew-free synchronous clock over the whole chip. GALS systems need to guarantee that synchronization failures between two different clock domains (metastability) will not occur. Many GALS interfaces have been proposed in the literature (BEIGNÉ, 2006), (SHEIBANYRAD, 2007), (PANADES, 2007), (THONNART, 2009), (STRANO, 2011) to cope with these problems and the majority of them are based on the use of asynchronous FIFOs with dual clock domains (PANADES, 2007), (STRANO, 2011).

The use of GALS interfaces is extremely important, mainly when a hierarchical network is used. The proposed hierarchical NoC allows each cluster to operate at the ideal

frequency. Moreover, as the clusters are heterogeneous, each cluster will have a different maximum operating frequency. To allow each cluster to operate at its ideal frequency, the bridges have a synchronizer based on the proposal of (PANADES, 2007) that uses a bi-synchronous FIFO to synchronize the clocks between the two regions.

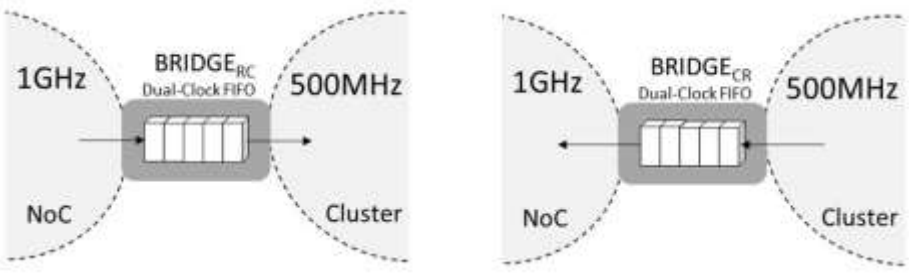
Table 3.2 presents the synthesis results for 65nm process technology for the implementation of the bi-synchronous FIFO according to (PANADES, 2007), considering two synchronization process: reading and writing. Figure 3.14 illustrates an example of the of the Dual-Clock FIFOs operating in the hierarchical NoC.

Table 3.2 - Synthesis results for the synchronizer module considering two frequency possibilities.

	Area (um²)	**Power (uW)**	**Frequencies**
$Bridge_{RC}$ (Fig. 3.14 (a))	1384	518	Read Freq.= 1GHz Write Freq. = 500MHz
$Bridge_{CR}$ (Fig. 3.14 (b))	1316	702	Read Freq. = 500GHz Write Freq. = 1GHz

Figure 3.14 - Example of the use of a Dual-clock FIFO for reading and writing from NoC and cluster with different operating frequency. (a) $Bridge_{RC}$, (b)$Bridge_{CR}$.

(a) (b)



Source: elaborated by the author.

3.3 ASHiNoC tool

As the number of ports of the clusters can be defined according to the requirements of an application, a tool with multiple objectives was developed in order to obtain the optimal HiCIT topology to be application-specific. In the context of application-specific NoC design, both the knowledge of communication patterns and core dimensions should be exploited for optimization purposes, avoiding the design for either the worst case or average case, which means either waste of area or possible communication bottleneck respectively. In fact, the knowledge of the communication patterns helps to find the best core allocations, reducing the number of intermediate network hops and consequently reducing the network power and

latency. On the other hand, knowledge of the core size can be used to customize the topology taking into consideration that not all the router-to-router links will have the same size.

A hierarchical NoC topology brings many advantages for complex MPSoCs since it can exploit the communication locality of the system, while still maintaining the NoC advantages. On top of this, the hierarchical NoC topology has proved not only to reduce the number of hops when compared to a regular packet-switched topology, but also is able to provide a suitable bandwidth, power and QoS results. In this context, the target Application-Specific Hierarchical NoC (ASHiNoC) tool is proposed to obtain the optimal HiCIT topology which is application specific (MATOS, 2011a). The proposed tool was developed in partnership with the Dipartimento di Elettronica e Informazione - Politecnico di Milano.

The advantage introduced through the usage of hierarchical interconnection topology for heterogeneous systems, can only be exploited by automatic tools for the design space exploration during the NoC synthesis. In fact, the large design space introduced by the mapping and clustering decisions, together with NoC and XBars architecture configurations is not possible to be faced manually. The application-specific network-on-chip synthesis which this proposal is facing is carried out by selecting the NoC parameters (such as frequency and router buffer depth), the topology and by the mapping of application cores into the network nodes (core-to-node mapping). In particular for the target Hierarchical NoC, the synthesis is also involved in the generation of the local Xbar-based cluster interconnections. The synthesis step is performed by taking into account estimated performance, power consumption and latency, considering floorplanning information.

Many research groups proposed tools to obtain an application-specific NoC topology. In (CHAN, 2008) an iterative methodology was presented to obtain an energy efficient topology supporting point-to-point and packet-switched connections. Similarly in ASHiNoC tool, this considers the effects of the wire length according to the topology. The topology generation algorithm consists of two phases: a phase that uses a topology as an initial starting point, and a refinement phase where the possibilities of adding or removing the links are analyzed to compose a point-to-point or router connection.

ASHiNoC proposal is different in the sense that with clusters composed of crossbar components, it is possible to reuse some of the communication links. Mapping and topology customization were also presented in (GIANLUCA, 2007). In this the authors proposed four different approaches for mapping and topology, but in this case, they were applied to STNoC architecture. The methods of mapping are based on the orthogonalization concept. The topologies are derived from ring and spidergon topologies and this proposal considers routers

with only three input/output ports for the connection between the NoC routers and one port to connect the core. However, the topology exploration in that work is limited, since the proposal only concerns the STNoC architecture. Another proposal for an application-specific network-on-chip architecture was presented in (YAN, 2006). That work proposed two heuristic algorithms to examine the partition possibilities for communication flows. The first algorithm was called CLUSTER and it reduces the number of set partitions from a smaller subset of set partitions. The second algorithm was called DECOMPOSE and, in this case, it starts with a single cluster and creates new partitions for each iteration. The two approaches generate the network topology using a Rectilinear-Steiner-Tree based algorithm to evaluate the costs of each partitioned group. However, that proposal only evaluates the power results of the topologies found.

(MURALI, 2006) presented a method that considers the floorplan in the wiring complexity, in the NoC topology synthesis process. The proposed mechanism integrates the synthesis, generation, simulation and physical design processes. Firstly the tool considers a set of constraints defined by the user. In the next step the NoC architecture is synthesized and a topology that best adheres to the design constraints is chosen, ensuring the required bandwidth of the application. After obtaining the synthesized topology, the floorplanning information is extracted. For the cluster partition the authors used the min-cut strategy to group the cores. The connection among clusters was made using a path allocation algorithm.

In (YU, 2010), a floorplan-aware method focusing to minimize the power consumption was presented. The authors used a partition driven floorplan to obtain the physical information. The switches and network interfaces were defined using heuristic method and a min-cost max-flow algorithm, respectively. The difference in this proposal is that it considers the network interfaces in the generation of the topology. The alternative to minimizing power consumption is achieved through the use of incremental path allocation. Another similar proposal is found in (LAI, 2011). That work tackles the same problem, but uses a genetic algorithm to synthesize application-specific NoC topology. However, the majority of these proposals consider clusters connected by switches with buffers and many other controls that increase the area and power dissipation. The differential aspect of the proposed tool is that it considers the HiCIT, topology which was not proposed by any of the researchers.

This work presents an automatic application-specific framework exploiting hierarchical interconnection, based on floorplanning information, considering core clustering, NoC mapping and interconnection synthesis problems. The goals of the ASHiNoC tool are:

- Proposing an automatic Design Space Exploration (DSE) framework for designing hierarchical network topologies based on floorplanning considerations;
- Obtaining an application-specific optimization in terms of area, power and latency of the hierarchical interconnection infrastructure, while maintaining communication requirements, considering both core mapping and clustering problems and architectural parameters tuning.

3.3.1 Problem Formulation

To define the problem of designing an Application Specific Hierarchical NoC, it is necessary to introduce the following concepts:

- *Core Graph* - used to describe the target application;
- *NoC Topology Graph* - used in this approach to describe the Global NoC level and the Core-to-Node mapping function used to correlate the two previous graphs.

The *Core Graph* is a directed graph $G(V,E)$ where V is the set of PEs belonging to the target System-on-chip and E is the set of edges representing the communication between the PEs $vi \in V$ and $vj \in V$. The weight of the edge $\varepsilon i,j \in E$, denoted by $commi,j$, represents the bandwidth of the direct communication from vi to vj.

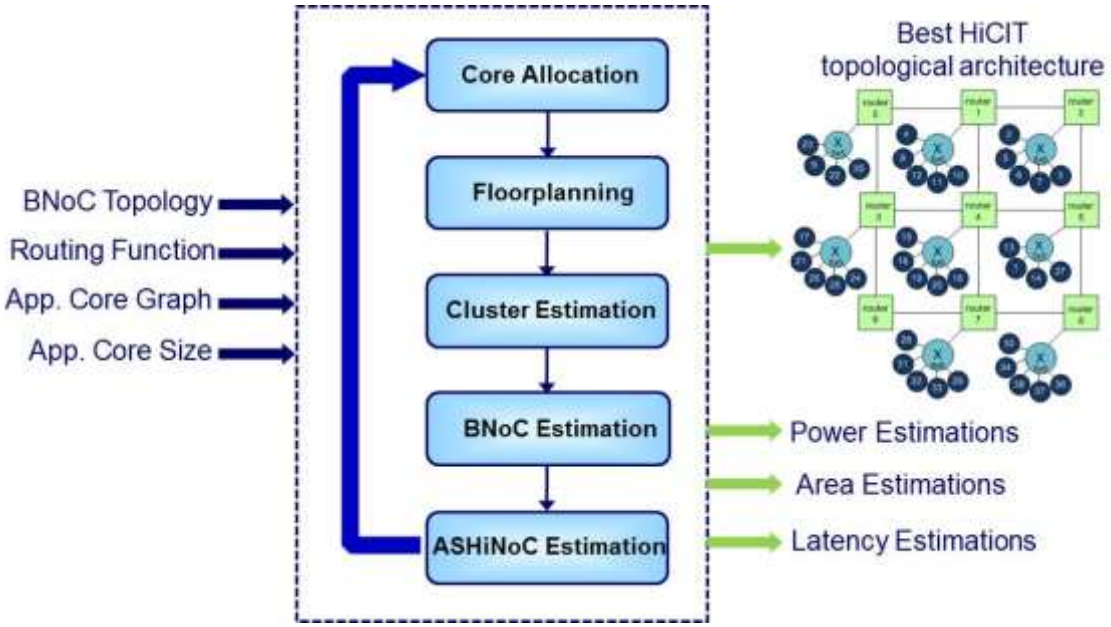
The *NoC Topology Graph* is a directed graph $P(U, F)$ where U is the set of network nodes and F is the set of directed edges (ui,uj) representing an existing link between the network nodes $ui \in U$ and $uj \in U$. Each edge $fi,j \in F$ has a weight bwi,j which represents the bandwidth available across it fi,j . The *Core-to-Node mapping function* $M : V \rightarrow U$ is defined as the set Core-to-Node mappings (vi, uj), representing the Core $vi \in V$ mapped to the network node $uj \in U$. The set of possible mappings $M(P, G)$ depends on a given network topology graph P and a Core Graph G.

Despite neither the *NoC Topology Graph* nor the Core-to-Node mapping function explicitly managing the cluster hierarchy, it is possible to manage the hierarchy in the mapping problem formulation, considering two (or more) cores mapped into the same network node as part of the same cluster, interconnected by a crossbar-based interconnection. Moreover, to evaluate the performance of each mapping and of the related Application-Specific Hierarchical NoC, including the floorplanning information, an enhanced version of the *Core Graph* was considered, where each core was annotated with the equivalent area and aspect-ratio.

3.3.2 Overview of the application specific methodology

The proposed tool flow is shown in Figure 3.15. Its input is the definition of the target topology in terms of number of routers and pattern (e.g. mesh, ring), the routing functions associated are with the topology and the enhanced version of the Core Graph (also annotated with the core dimensions). From these parameters it generates output as the optimal HiCIT architecture together with the power/area/latency evaluations. It is composed of 5 main steps completed iteratively: core allocation, floorplanning, cluster estimation, backbone NoC (BNoC) estimation and hierarchical NoC estimation.

Figure 3.15 - ASHiNoC analyzer framework.



Source: elaborated by the author.

3.3.1.1.Core Allocation

This is the step that is in charge of the cluster generation, each core of the Core Graph G(V,E) is mapped on a specific cluster. The core graph relays the behavior of an application through the core communication rates and how they are interconnected, identifying the regions of communications. The core allocation of this tool has been implemented by using the NSGA-II multi-objective genetic algorithm (DEB, 2002) since the design space of the target problem gives results too vast to be analyzed exhaustively. The adopted chromosome structure is composed of a set of genes equals the number of cores in the Core Graph. Each gene identifies the cluster where the related core has been mapped. This chromosome

structure has the possibility to be extended by also encoding additional genes for other architectural parameters related to the hierarchical NoC (e.g. Data-path width, frequency).

3.3.1.2. Floorplanning

This step analyzes the core-to-cluster assignment, performing the floorplan estimation for each cluster and for the backbone NoC. The floorplan has been derived using the Hotfloorplan tool (HOTSPOT). It uses a two dimensional layout, where each core is represented by a rectangular region, with size and aspect ratios specified in the input Core Graph. Each local crossbar and backbone router (together with its related Network Interface) are represented by a square with size dependent on the number of cores connected and the backbone NoC topology respectively. The area model for both local crossbars and NoC routers has been derived using ORION 2.0 (KAHNG, 2009). ORION 2.0 is an enhanced NoC power and area simulator used by various researchers offering significant accuracy. The generated floorplan is based on the topology of the entire hierarchical network, by minimizing a linear combination of the silicon area for the entire floorplan and the distance between interconnected resources.

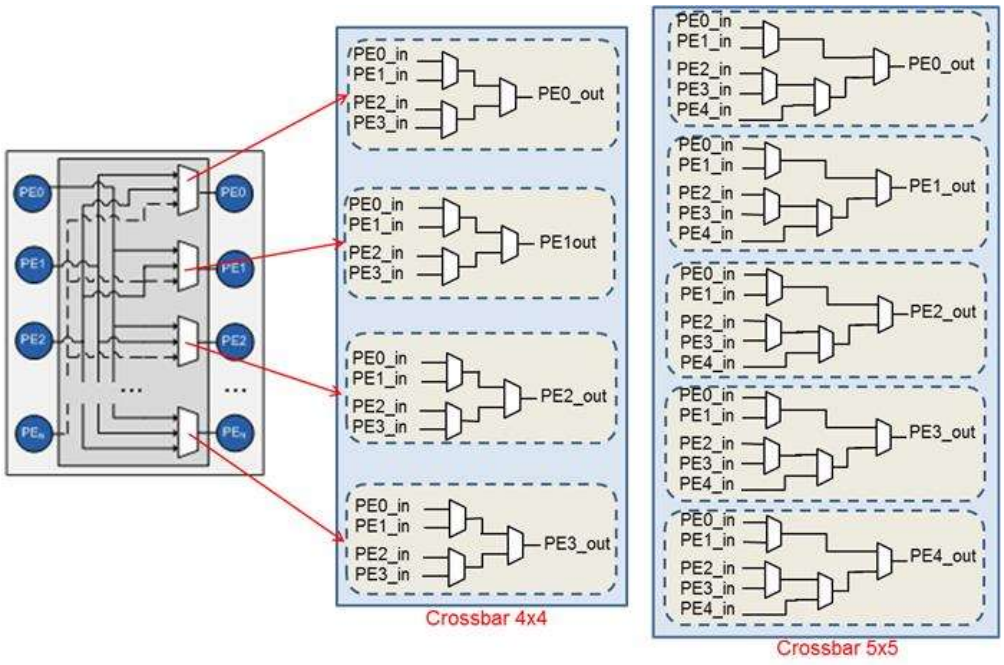
3.3.1.3.Cluster Estimation

This step uses the previous floorplanning phase to design each XBar-based interconnection and estimate its performance in terms of maximum working frequency, average power and area. First of all, each cluster floorplan is used to estimate the length of the links attached to the local Xbars. Next, the estimated links length is processed by an in house model that is able to extract the maximum working frequency of each cluster, depending on the wire latency and crossbar size. The model was obtained from Hspice results for a 65nm process technology. A distributed RLC-π model (SAKURA, 1983) was used, which emulates the electric wire behavior. The values of resistance and capacitance for this model, for 65nm technology were obtained from (PTM, 2013). For these experiments, the multiplexer gates that compose the Xbar were connected in a binary tree, as illustrated in Figure 3.16. In these results, a tree of 2:1 MUX (multiplexer) presented smaller latency than a 2:1 MUX and 4:1 MUX composition. As can be seen, increasing the crossbar in one port, one 2:1 MUX is inserted in each set of multiplexers required for each output port (see example of Figure 3.16 for radix-4 and radix5). In order to estimate the crossbar delay, the critical path was verified, as shown in Figure 3.17. The estimation error for the cluster was approximately +7% and -7% for the latency in the worst cases analyzed when compared to the measured results (from

Xbar radix-3 to Xbar radix-16). The estimation error for the mesh level is the same of the ORION 2.0 tool ((KAHNG, 2009).

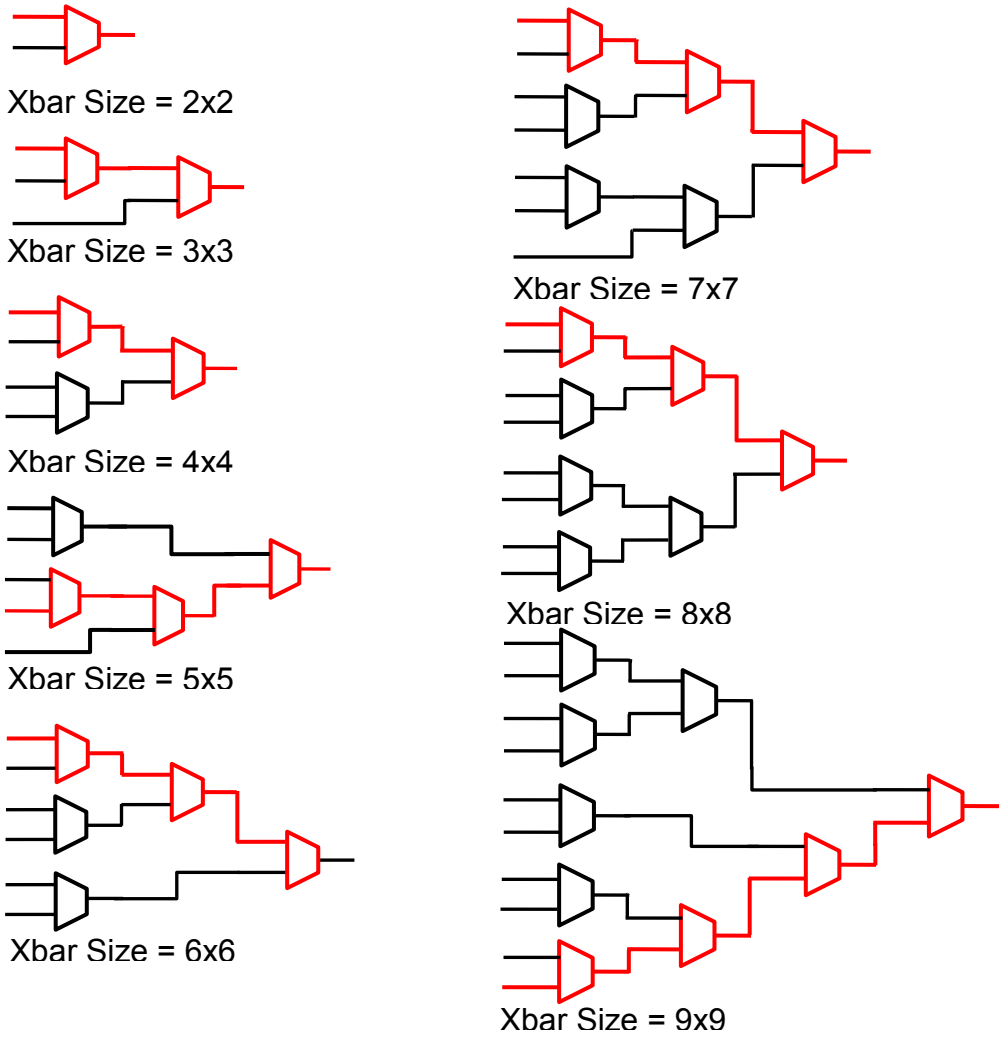
Finally, area and average power values for both crossbars and links have been derived using ORION 2.0 (KAHNG, 2009) taking into consideration the estimated links length, the crossbars size in terms of ports, the data-path width and the communication traffic. This step is concluded with the verification that each crossbar link is able to sustain the allocated bandwidth. Otherwise, a bandwidth constraint violation is reported. The number of violated bandwidth constraints is used by the exploration algorithm to discriminate between good and bad designs and the selection of the next mapping function to be analyzed.

Figure 3.16 - Number of crossbar 2:1 MUXES according to the number of input ports.



Source: elaborated by the author.

Figure 3.17 - Critical path for each crossbar size defined to estimate the crossbar delay.



Source: elaborated by the author.

3.3.1.4. Global level Estimation

Once the per cluster analysis has been concluded, this next step estimates the backbone network performance in terms of maximum sustainable frequency, area and average power consumed by each router and by each router-to-router link. This estimation step is made by considering the BNoC topology, the intercluster-level floorplan for deriving the length of the network links and again using ORION 2.0 (KAHNG, 2009) for the technology parameters. As previously done for the local crossbar, this step is concluded with the verification that each backbone NoC link is able to sustain the allocated bandwidth.

3.3.1.5. Hierarchical NoC Estimation

Finally, the last step of the methodology is used to put all the metric derived for the local crossbars and the backbone NoC together. The area and average power of the ASHiNoC is obtained from the sum of crossbars and BNoC values, while for the average latency the

weighted sum for each edge in the communication graph of the ideal latency (IL) was used, and can be expressed in equation 2:

$$IL = CL_s + \frac{hops\ x\ cycles}{freq_{BNoC}} + CL_d \quad (2)$$

where CL_s and CL_d are the latency due to the source and destination cluster respectively, $freq_{BNoC}$ is the backbone NoC working frequency, while hops and cycles are the number of intermediate hops and network cycles respectively, introduced by each hop.

3.4 Results

In this section different results for the HiCIT architecture will be presented. Initially, the estimation results obtained with the ASHiNoC tool in terms of area, power and latency will be demonstrated. These results are important in understanding the exploitation possibility and the different results obtained when some parameters are changed. Posteriorly, the synthesis results for different applications and a trend of the synthesis results for MPSoCs with different number of nodes will be shown.

In order to validate the presented solution, as the proposed architecture focuses on heterogeneous systems, where the use of hierarchy present advantages, the applications used to analyze the results need to present an unbalanced communication bandwidth, since the cores are heterogeneous in the communication rates and size. Moreover, as previously commented in this section the physical core area is very relevant in the floorplanning design, since these features involve the interconnection costs and power consumption. In this manner, this work uses some standard benchmarks referenced in the literature to validate the proposed strategies. The first application analyzed is a video decoder called TVOPD (Triple Video Object Plane Decoder) (MURALI, 2009) whose basic components were tripled to increase performance. This application presents high heterogeneity, composed of 3 independent decoders, with 12 PEs each, running in parallel. TVOPD presents two memories shared for each decoder, playing the role of input and output buffers. As in (MURALLI, 2006), the values related to the core areas of TVOPD were obtained for 130 nanometer process technology and the results in this research were verified for smaller technologies (65nm) using a simple method for technologic scaling. In this case, as the relevance of this analysis is considering the heterogeneity of the applications, the scaling of the core area considers that the area decreases by the square of the scale difference. In this case, the area was divided by 4. The core area scaling for TVOPD is presented in Table 3.3.

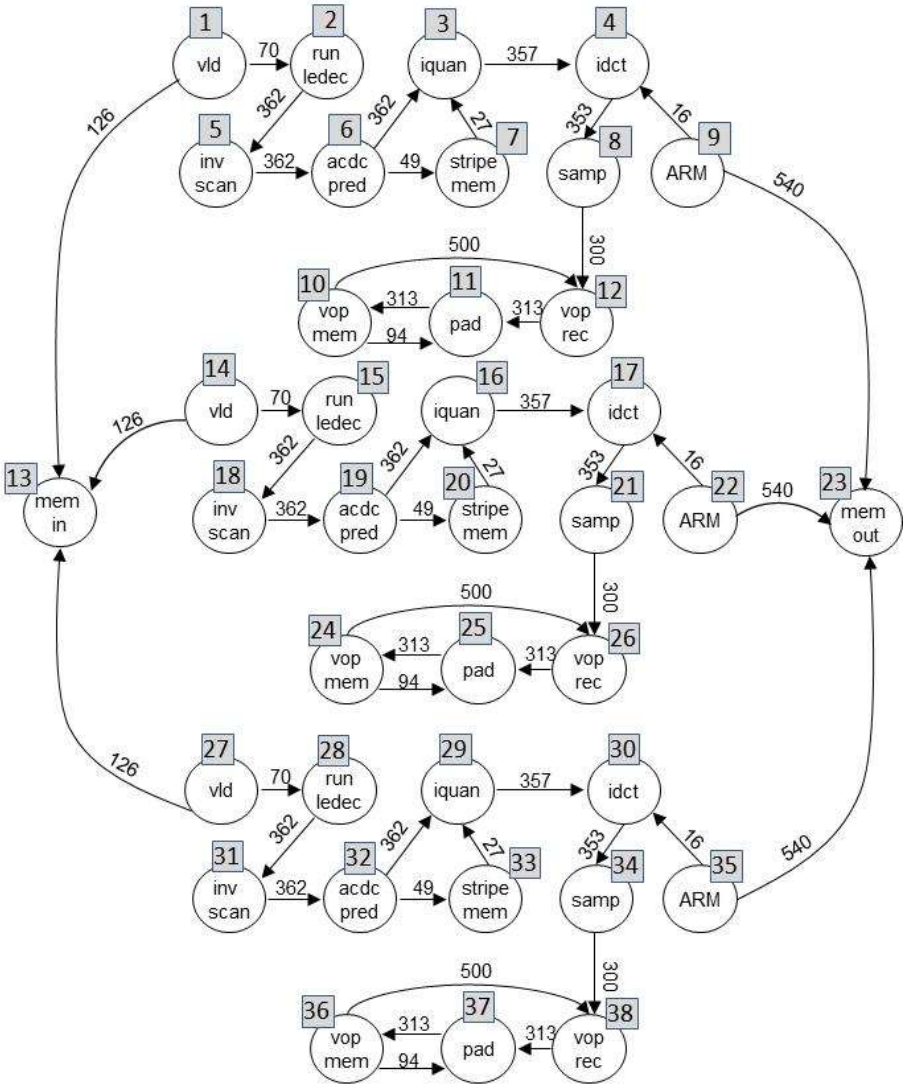
Table 3.3 - Physical information of TVOPD cores.

Cores	130 nm			65 nm
	length (mm)	width (mm)	area (mm²)	area (mm²)
RLD	1.2135	1.1837	1.4363	0.3591
ISCAN	1.2135	1.1802	1.4321	0.3580
ARM	0.9844	0.9775	0.9623	0.2405
VLD	1.1884	1.1801	1.4025	0.3506
IDCT	1.1884	1.1801	1.4025	0.3506
PAD	1.1884	1.1908	1.4152	0.3538
UPSAMP	1.1884	1.1908	1.4152	0.3538
ACDC	1.1741	1.1801	1.3856	0.3464
SMEM	1.9831	1.9693	3.9054	0.9764
VOPM	1.1884	1.1908	1.4152	0.3538
VOPR	1.1991	1.1908	1.4280	0.3570
IQUANT	1.1991	1.1659	1.3982	0.3495
MEM_IN	1.9831	1.9693	3.9054	0.9764
MEM_OUT	1.9831	1.9693	3.9054	0.9764

3.4.1 ASHiNoC tool results

In this section, the experimental results applying the proposed ASHiNoC methodology will be shown. In these experiments, the case study was the TVOPD application, composed of 38 cores. The communication core graph structure is presented in Figure 3.18 and has been derived from (MURALI, 2009). The communication rates identified in the Figure 3.18 are represented in MB/s.

Figure 3.18 - TVOPD core graph.



Source: MURALI (2009).

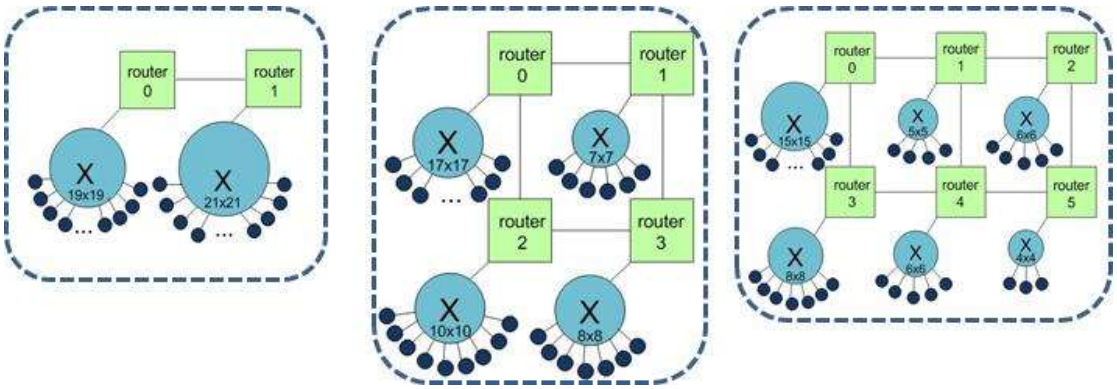
Regarding the network evaluation, synthesis and estimation values were obtained using 65nm process technology, considering the routers' architecture as having an input queue depth equals 4 slots. The larger buffer depth, the smaller network contention and larger power consumption. In this manner, to appropriately verify the impact of the power consumption of the proposed architectures, one was defined a small FIFO depth for the router input channels. Because of this, in the majority of experiments of this work, buffer depth is equal to 4. The analyzed design space versus the target use case is composed of all the possible configurations, combining the following parameters:

• 38 core cluster assignment and network mapping parameters, varying from 0 to Max BNoC routers (one for each core);

• 1 number of the mesh backbone NoC routers parameter, varying in the following set 2, 4, 6, 9, 12, 16 but keeping the number of cores in the system (38 in the TVOPD), as the example of Figure 3.19 (all cases with the same cores) ;

• 1 BNoC frequency parameter, with the following variations 250MHz, 500MHz, 750MHz, 1000MHz.

Figure 3.19 - Example of HiCIT architectures varying the crossbar size for the same application.



Source: elaborated by the author.

The resulting design space is composed of more than 1046 architectural configurations, and it is not feasible to be explored without the use of an automatic design space exploration technique. For the application-specific hierarchical NoC generation, it formalizes the multi-objective optimization problem, as presented in (3), and is subject to the respect of the application communication bandwidth.

$$\min_{x \in X} \begin{bmatrix} area^{ASHiNoC}(x) \\ average\_power^{ASHiNoC}(x) \\ average\_latency^{ASHiNoC}(x) \\ routers^{BNoC}(x) \end{bmatrix} \quad (3)$$

In particular, the minimization problem faced in this use case consists of four objective functions. Three of the objective functions are related to the entire HiCIT interconnection: the area, the average power consumption and the average latency. The fourth refers to the Backbone NoC component of the HiCIT interconnection and in particular to the number of routers. The minimization problem does not have any constraints, except supporting the QoS requirements of the TVOPD application, the resulting design configurations should be able to support the data traffic presented in Figure 3.18, so, in this work, QoS guarantee means that the required communication bandwidth is accomplished. To help the system's architect choose between the large number of feasible solutions composing the Pareto front, and better analyze the impact of the hierarchy in the different design configurations, it was decided first

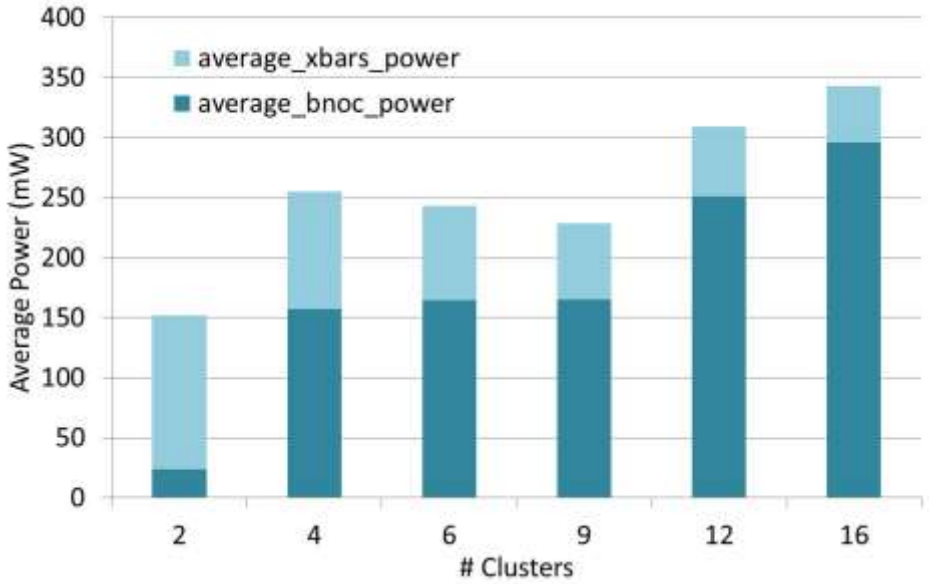
to cluster the Pareto solution with respect to the number of routers in the Backbone NoC and then to select the best solution for each cluster by using a decision-making mechanism based on the following product, equation 4:

$$area(x) \times avg_{power(x)} \times avg\_latency(x) \quad (4)$$

The results of all tool steps can be found in Figure 3.20, Figure 3.21 and Figure 3.22, which plot the interconnection Average Power and Area (Figure 3.20 and 3.21, respectively) and the Average Latency and Average Hops for the best configurations found by varying the number of routers (Figure 3.22). As the optimal solution for each number of clusters was obtained, comparisons between the solutions may not have a direct correlation, once for each solution the core mapping can be completely different, but it is possible to observe a trend from the results.

Figure 3.20 shows the HiCIT average power, splitting the Xbars and Backbone NoC power. Except for a small reduction passing from 4 to 9, the global power trend increases with the number of BNoC nodes. While the backbone NoC power component, which is the main responsible of the global trend, increases with the number of nodes, the opposite happens for the Xbar component. This is mainly due to the increment of data traffic that passes from inter-cluster to intra-cluster communications, removing the pressure from the local Xbars and also due to the increase in the number of routers in the topology. Passing from 4 to 9 crossbar switches, an almost constant BNoC power together with a reduction in the Xbars' components causes a global reduction in the HiCIT power.

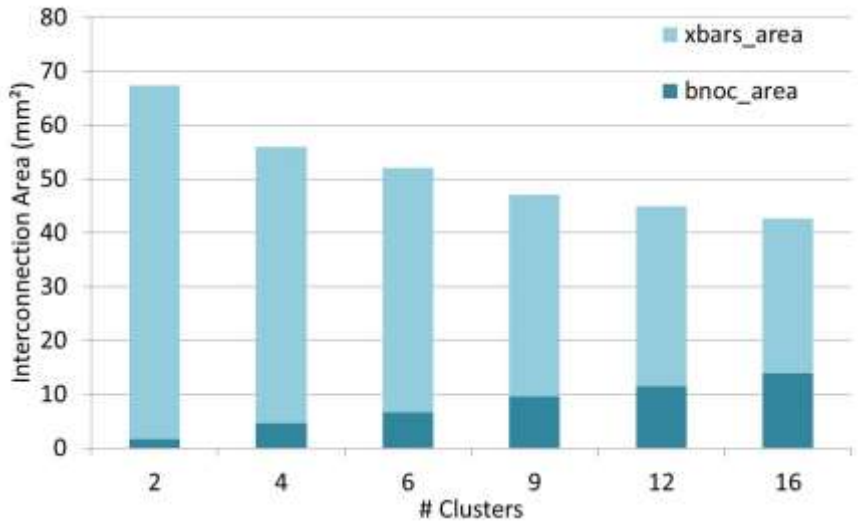
Figure 3.20 - HiCIT interconnection average power for the best configuration, varying the number of routers on the global level.



Source: elaborated by the author.

Figure 3.21 shows the ASHINoC area split into the Xbar and Backbone NoC components (mesh routers). The global area trend result in an ASHINoC area reduction as the number of BNoC nodes increases. In fact, large clusters (found when the number is small) require longer local interconnections relative to smaller clusters (found when the number is large) and therefore, larger areas. For the target case study, the area overhead, introduced by the additional BNoC routers, is reduced with respect to the advantage of reducing the cluster dimension, leading, to the observed area reduction trend.

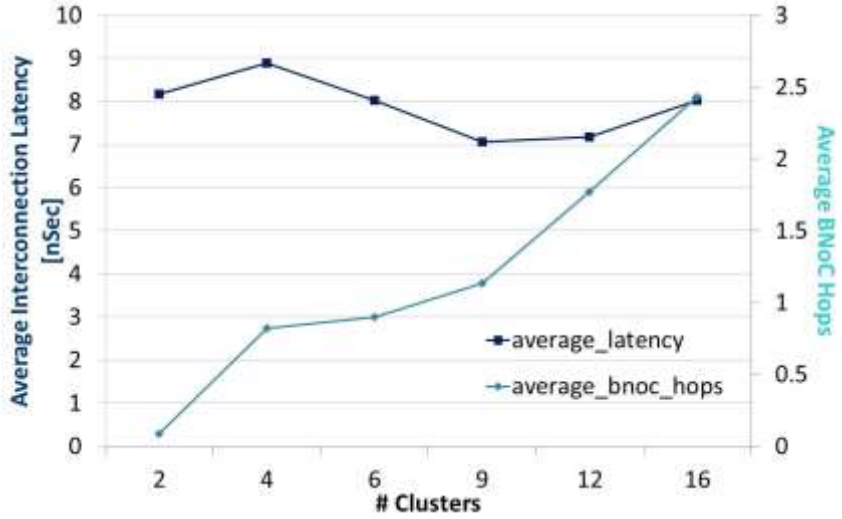
Figure 3.21- HiCIT interconnection area and for the best configuration, varying the number of routers on the global level.



Source: elaborated by the author.

Figure 3.22 shows the ASHINoC average latency and BNoC hops. It can be seen that as it moves from 2 to 16, the average number of hops increases, mainly due to the increase in data traffic that passes from intra-cluster to inter-cluster communications, and due to the increment of the average network distance (DALLY, 2003). This trend, which was expected, is not entirely reflected into the average interconnection latency, since the average hops metric masks entirely the impact of the clusters local frequency. In fact, larger clusters present a lower working frequency with respect to smaller clusters (considering the same area for the cores) also meaning a larger latency for all the intra-cluster communications. For the target case study, the average interconnection latency presents a trade-off with 9 BNoC nodes. For values smaller than 9 nodes, the increment of the working frequency is counterbalanced by the increment in the network latency, whereas for values larger than 9 the advantage of having smaller cluster are not more evident.

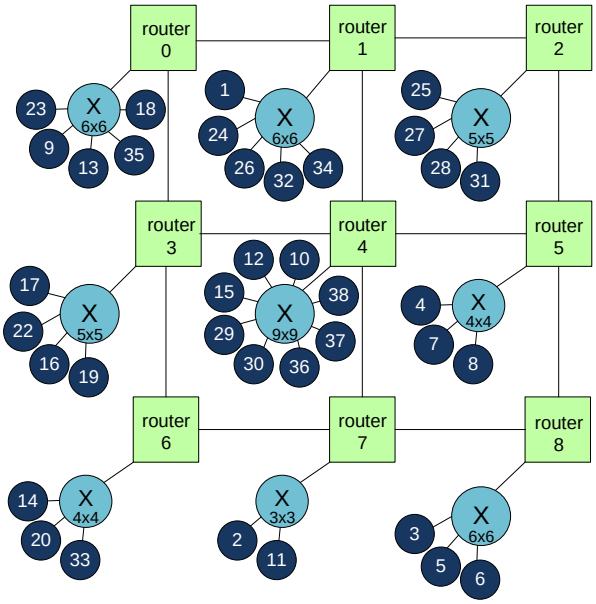
Figure 3.22 - HiCIT interconnection average Latency for the best configuration, varying the number of routers on the global level.



Source: elaborated by the author.

In addition to this, the best configuration among all the Pareto solutions using the same decision-making criteria presented in equation 4 for the TVOPD benchmark, is shown in Figure 3.23. Table 3.4 outlines the results of this final solution. As can also be noted from Figure 3.20, this solution is characterized by a higher power contribution from the Backbone NoC. As for the clusters organization, this solution is able to generate well balanced clusters in terms of the number of nodes (except for cluster 4) able to sustain a working frequency equals, in most of the cases, 500MHz.

Figure 3.23 - The Optimal HiCIT topology for TVOPD with 9 clusters.



Source: elaborated by the author.

Table 3.4 - Results for the best configuration for TVOPD benchmark.

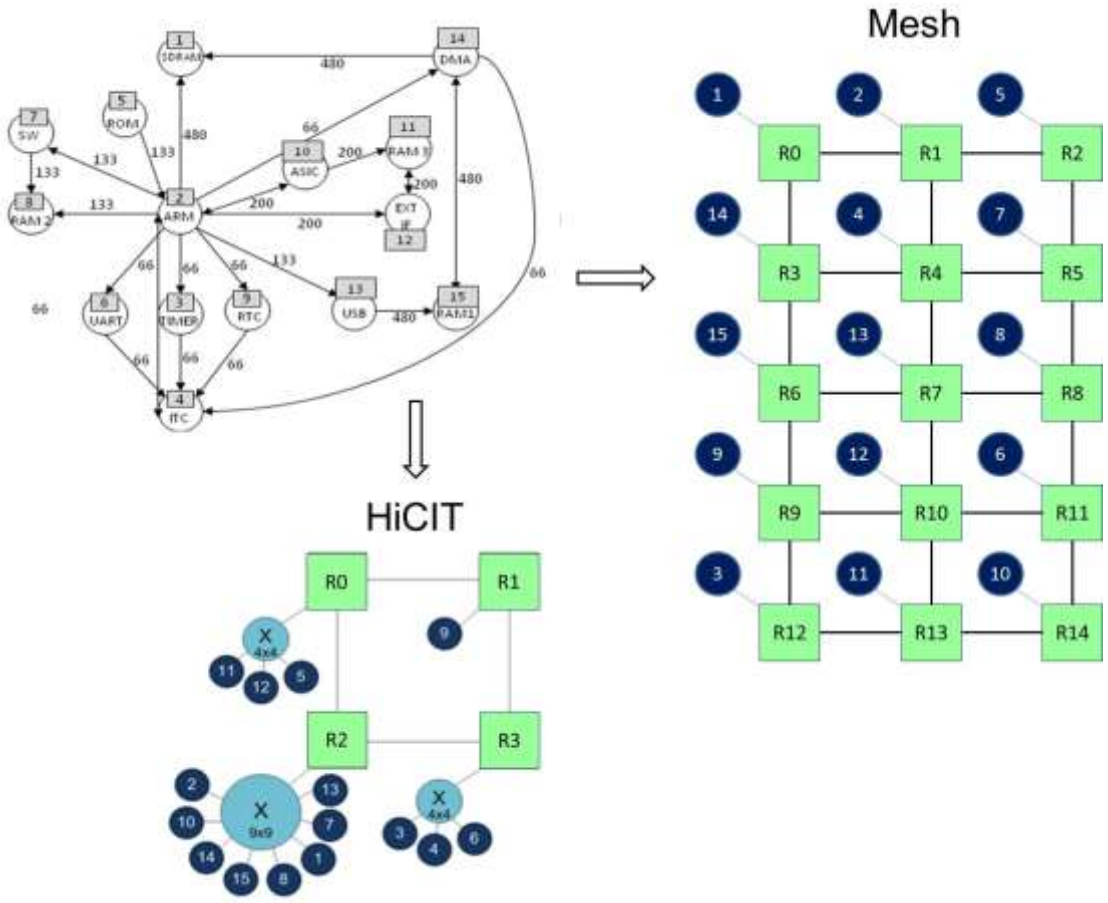
Backbone NoC					
Topology	Routers	Frequency (MHz)	Power (mW)	Area (mm^2)	
Mesh	9	500	165.36	9.62	
Xbar-based Clusters					
Cluster	Cores	Frequency (MHz)	Power (mW)	Area (mm^2)	Core Name
0	5	500	10.12	5.34	ARM0, ARM2, ISCAN1, MEMOUT, MEMIN
1	5	500	9.57	4.87	VLS0, VOPM1, VOPR1, UPSAMP2, ACDC2
2	4	500	4.49	3.58	PAD1, RLD2, ISCAN2, VLD2
3	4	500	6.21	3.51	ARM1, IDCT1, ACDC1, IQUANT1
4	8	250	18.97	9.18	VOPM0, VOPR0, RLD1, IDCT2, PAD2, VOPM2, VOPR2, IQUANT2
5	4	500	5.91	3.93	IDCT0, UPSAMP0, SMEM0, UPSAMP1
6	3	500	1.78	3.04	VLD1, SMEM1, SMEM2
7	2	1000	1.95	1.53	RLD0, PAD0
8	3	750	4.03	2.48	ISCAN0, ACDC0, IQUANT0

Besides the TVOPD, two other benchmarks were evaluated: TMPEG4 (triple MPEG4 (BERTOZZI, 2005) in a similar construction to the TVOPD), and the NCS (Network Communication Subsystem) (TINO, 2010). The same algorithm was used to obtain an optimal mapping for HiCIT and mesh topologies. Optimal mapping was found with the ASHINoC tool for these benchmarks and the results for these topologies were compared to a flat mesh NoC. TMPEG4 decoder can be found in many embedded applications. It presents memories with a high communication demand. In the TMPEG4, the MPEG4 module was replicated, but in this case, the SRAM1 was shared with the cores belonging to other modules instead of adding another memory component (see Figure 3.25). NCS is used as an industrial network communication SoC subsystem. This SoC is responsible for supporting two TCP constraints: an encryption engine and a USB subsystem. In the encryption engine, the EXT_IF core fetches data and stores it back in the RAM. The EXT_IF block then fetches the

data for streaming data back. In the USB subsystem, the system receives data from the USB and transfers it to RAM1 where the DMA engine streams it to the SDRAM_IF for further streaming transmissions (TINO, 2011).

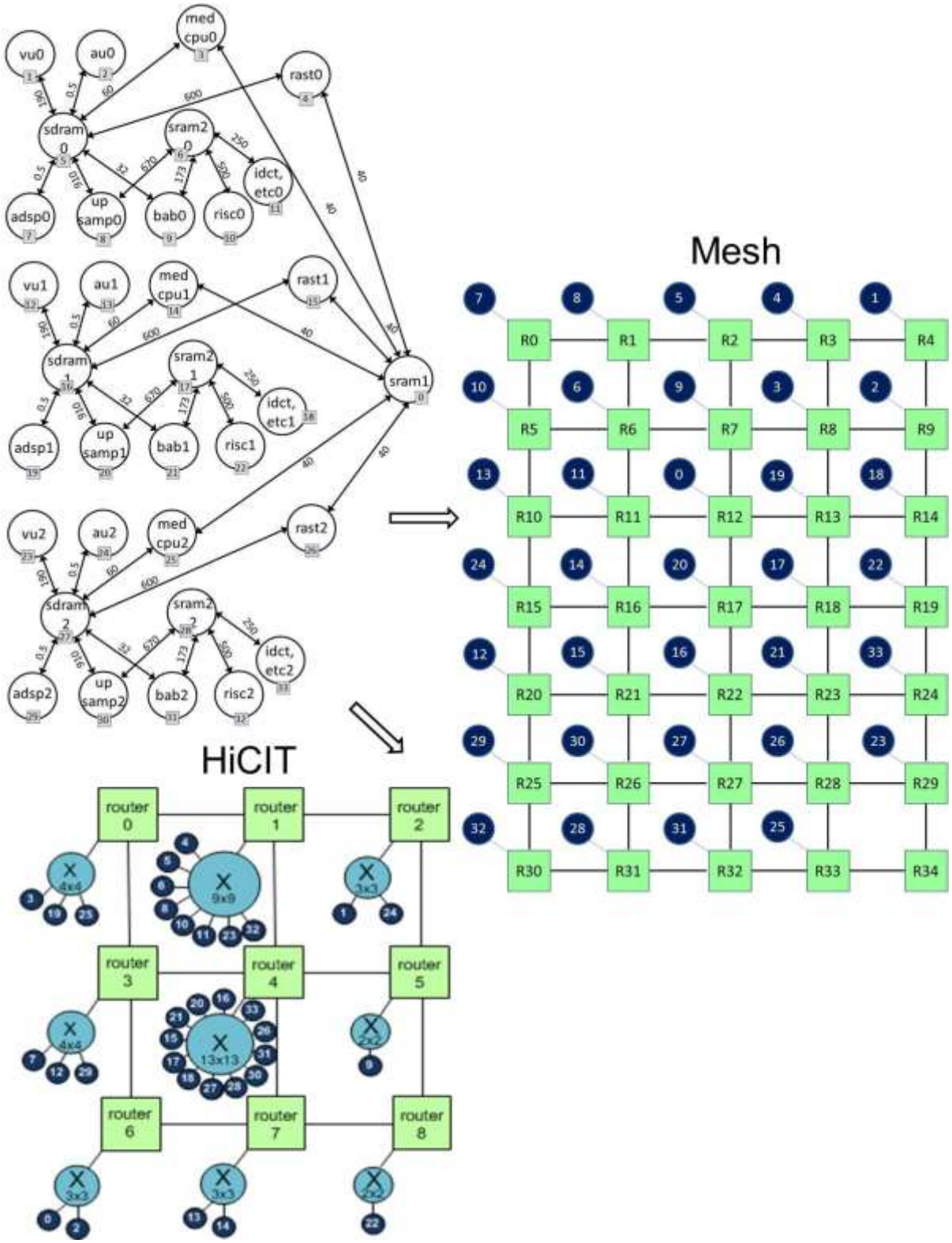
The core graphs for NCS and TMPEG4, and the corresponding mapping for HiCIT and mesh topologies are shown in Figure 3.24 and Figure 3.25, respectively. Both bandwidths shown on the edges of the core graphs represent the communication requirements in MB/s. Figure 3.26 presents the TVOPD mapping for mesh topology used in the comparison with HiCIT.

Figure 3.24 - Mesh and HiCIT topologies for NCS benchmark.



Source: elaborated by the author.

Figure 3.25 – Mesh and HiCIT topologies for TMPEG4 benchmark.

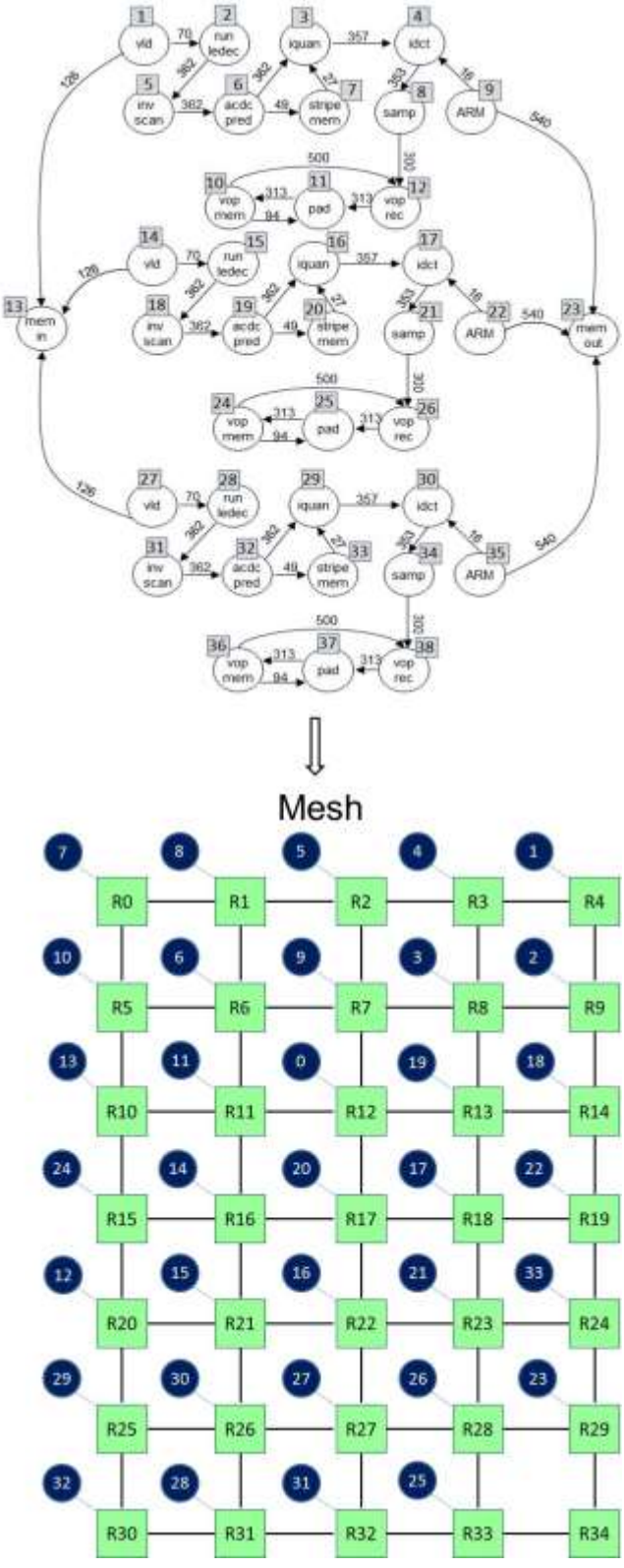


Source: elaborated by the author.

An input parameter of the tool is the topology and NoC size. In this manner, the cores are distributed according this definition. Although in some cases, it seems better to define other core distribution in the clusters, the tool algorithm can obtain better results for other

configurations, separating communicating cores in different clusters, as in the example of RTC and RISC, put alone in the router R1 and R5 of the NCS and TMPEG4 HiCIT topologies, respectively.

Figure 3.26 - Mesh topology for TVOPD benchmark.



Source: elaborated by the author.

As for TVOPD benchmark, the core sizes of NCS and TMPEG4 were informed as input parameters of the tool for obtaining the best topology. The core area of NCS for 65nm was obtained from (TINO, 2010) and their respective sizes are illustrated in Table 3.5. The core area of TMPEG4 was obtained from (LAI, 2010). As these areas of this benchmark were obtained for 100nm technology, the same simple scaling applied to the TVOPD was considered for TMPEG4 and their respective sizes are illustrated in Table 3.6.

Table 3.5 - Physical information of NCS cores.

Cores	length (mm)	width (mm)	area (mm²)
ARM	1.20	1.16	1.39
ROM	0.50	0.60	0.30
SW	0.71	0.24	0.17
RAM2	0.73	0.28	0.21
USB2	0.66	0.35	0.23
EXT-IF	0.70	0.34	0.23
ASIC1	0.85	1.15	0.99
SDRAM-IF	0.43	0.68	0.29
TIMER	0.16	0.30	0.05
UART	0.33	0.21	0.07
RTC	1.03	0.40	0.41
ITC	0.91	0.25	0.23
RAM1	0.91	0.59	0.54
DMA	0.90	0.27	0.24
RAM3	0.72	0.60	0.43

Table 3.6 - Physical information of TMPEG4 cores.

	100nm			65nm
Cores	lenght (mm)	width (mm)	area (mm²)	area (mm²)
ADSP	0.448649	0.487416	0.218679	0.09266
VU	0.154054	0.167366	0.025783	0.010925
BAB	0.486486	0.516779	0.251406	0.106528
UPSAMP	1.048649	1.036493	1.086917	0.460558
SDRAM	1.705405	1.688339	2.879302	1.220043
MCPU	1.513514	1.526846	2.310902	0.979196
SRAM2	0.467568	0.475671	0.222408	0.094241
RISC	1.372973	1.347735	1.850404	0.784069
AU	0.143243	0.143876	0.020609	0.008733
RAST	0.456757	0.455117	0.207878	0.088084
SRAM1	1.459459	1.468121	2.142663	0.907908
IDCT	0.418919	0.405201	0.169747	0.071926

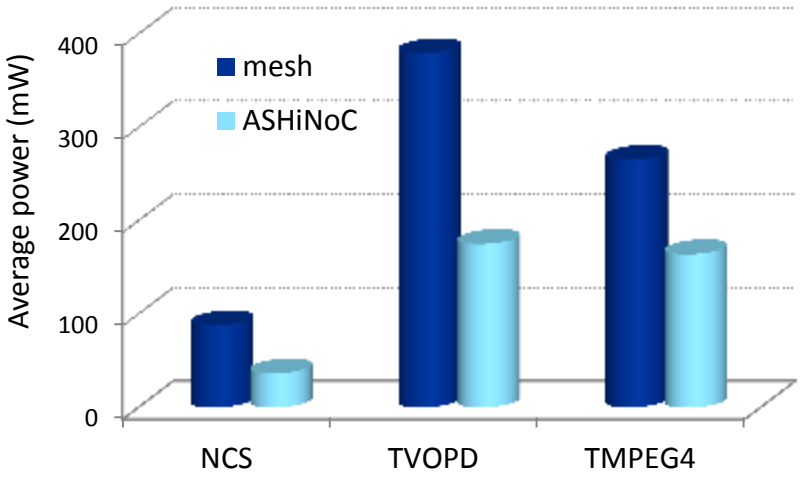
Figure 3.27 presents the comparison of the topologies in terms of average power, average network hops and average latency for these benchmarks. The mesh NoC was configured to operate at 1GHz and the links were set to 16 bits.

As one can see, the HiCIT topology allows a largely reduced average power, average latency and network hops when compared to a conventional mesh NoC that presents one core per router. The gains achieved with the HiCIT topology are confirmed by Table 3.7.

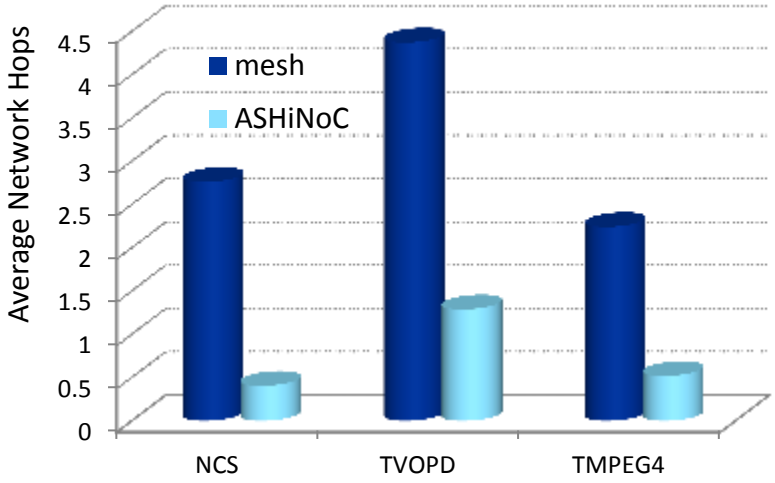
The power reduction reaches up to 58%, and the latency reduction up to 55%. The gains obtained with the hierarchical topology are still larger when compared to the average network hops. In these experiments, the average latency is for an ideal value, i.e., it considers the time required to send a message without contention in the network, while considering the total bandwidth of the communications that share the links. Results considering the hierarchical topology with the real latency (with contentions) will be presented in chapter 5.

Figure 3.27 - (a) Average power, (b) average network hops and (c) average latency for NCS, TVOPD and TMPEG4 in the HiCIT and mesh topologies.

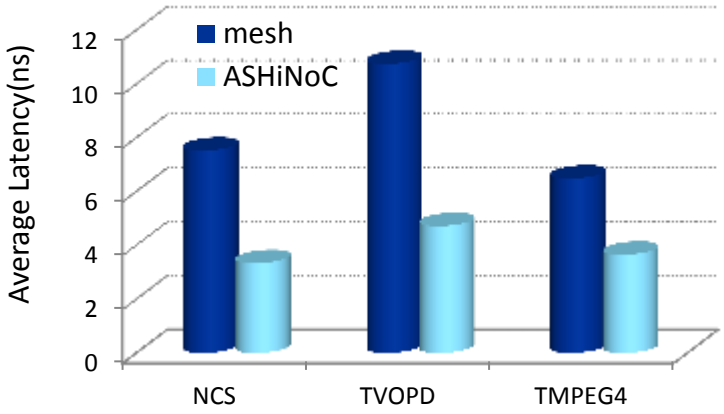
(a)



(b)



(c)



Source: elaborated by the author.

Table 3.7 - Reduction in power, NoC hops and latency with the use of HiCIT when compared to a flat mesh topology.

	Reduction in power	Reduction in NoC hops	Reduction in latency
NCS	58.5 %	85.3%	55.6%
TVOPD	53.9 %	70.5%	56.1%
TMPEG4	38.1 %	76.8%	43.7%

3.4.2 Synthesis results

The use of a high-level tool as ASHINoC allowed a quick verification of the advantages of the proposal. However, to obtain the final synthesis results, the power and area were obtained with Cadence Design Systems tools. Synthesis results present high computational costs, and as a result of this, a tool with a higher abstraction level is preferred when many experiments are needed.

Two configurations for HiCIT architecture were analyzed, one with clusters composed of 4 cores and the other composed of 8 cores. The NoC used in the comparison is the SoCIN (ZEFERINO, 2003). All results were compared to a 2D mesh with XY routing algorithm and wormhole switching. The data width of the links was set to 16 bits (the same as the crossbar links) and channels with 4-flit deep buffers.

A result exploration was verified for the HiCIT architecture composed of several cores. For these experiments, cores with an area equals 1mm² and operating frequency of 1GHz were defined. As previously commented, 1mm² is the area for an ARM1176 for 65nm technology. The operating frequency definition is a realistic value for current MPSoCs and widely used in the researches.

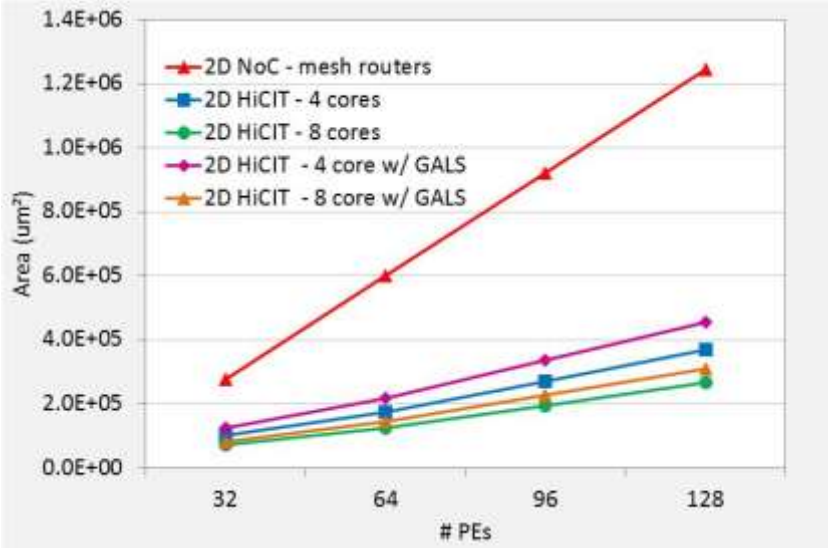
The area and power consumption of these experiments are presented in Figure 3.28. As can be seen, the area reduction for the HiCIT compared to a mesh topology is very high, reaching 75% for HiCIT with 4 cores per cluster and more than 80% for clusters with 8 cores. In this graph was also plotted the area results for HiCIT with GALS interfaces. GALS interfaces present a 16% increment in area for HiCIT with 8 cores and a 23% increment for HiCIT with 4 cores, but with the advantage to define distinct frequencies for each cluster and with this, reducing the power consumption. Even with this area increment, the total area is smaller than a mesh topology. As was presented, there are some advantages to including more cores in each cluster, but, according to Figure 3.7, this advantage is limited in few tens of cores for routers with small buffers.

With regards to power, it was possible to conclude with these last results, that for the parameters defined, the HiCIT power becomes larger than mesh NoC at around 100 cores.

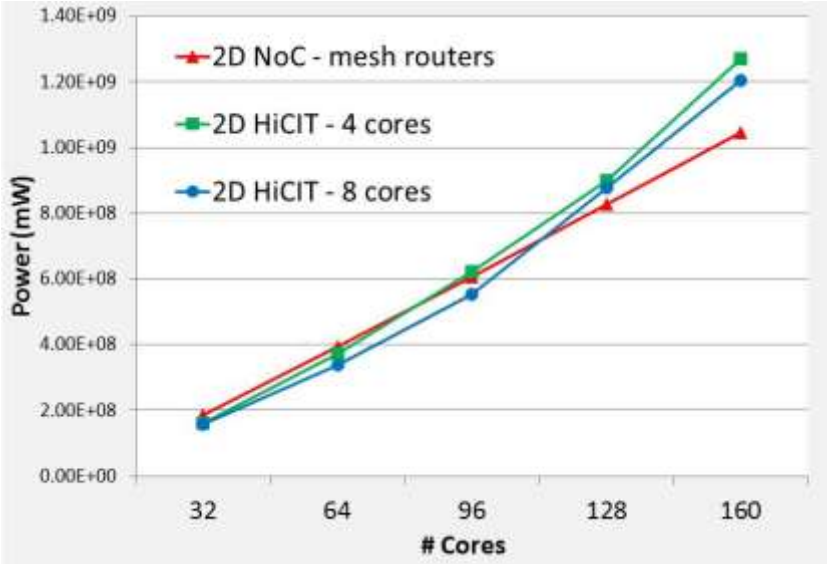
However, it is important to note that this comparison favors the mesh NoC since the same toggle rate was considered by the tool for the both topologies. In this manner, the wires in the HiCIT topology present a large power consumption (see Figure 3.29). As the power consumption is proportional to the switching activity and, being HiCIT formed by crossbars, its switching activity will be smaller than a router that present buffers. Crossbars, contrary to buffers, upon having defined the communication, set the multiplexers and maintain this state, at least until the end of the transmission.

Figure 3.28: (a) Area and (b) power consumption when the number of PEs increases.

(a)

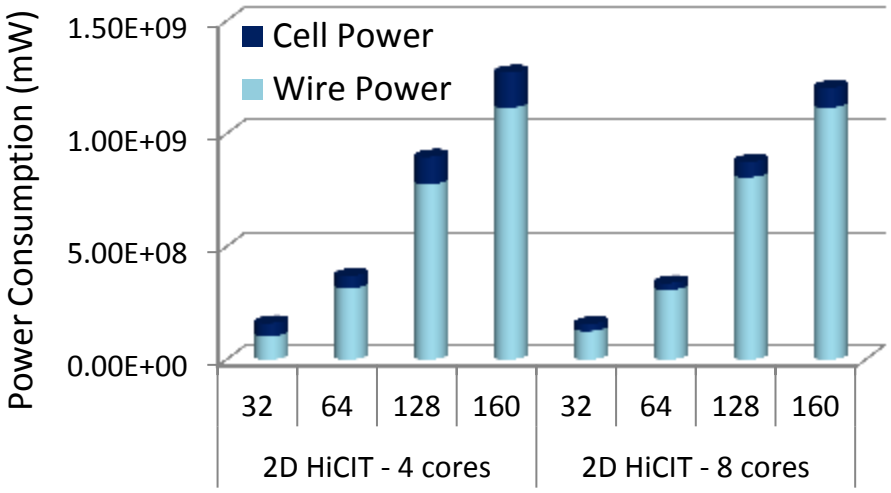


(b)



Source: elaborated by the author.

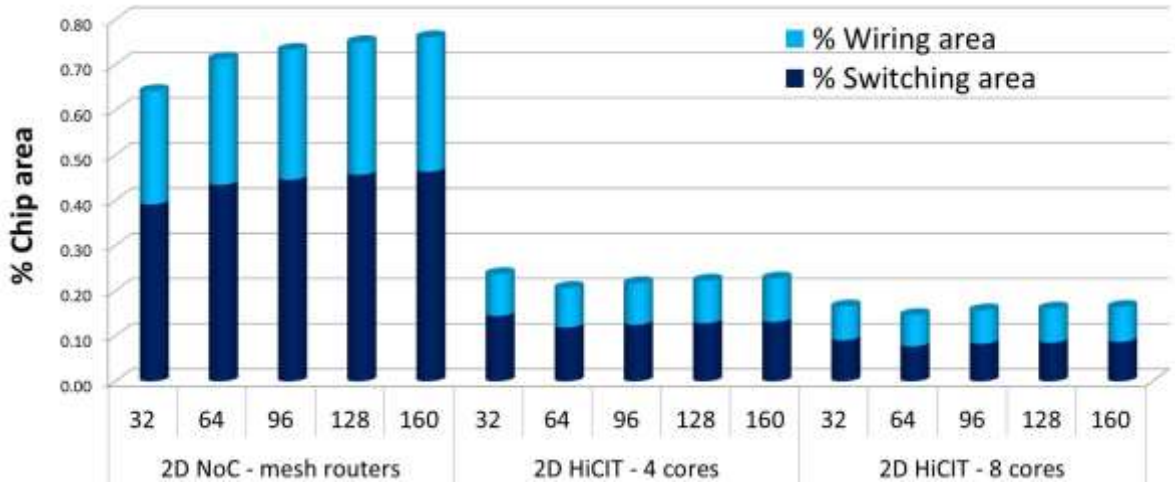
Figure 3.29 - HiCIT power consumption divided in wires and cells.



Source: elaborated by the author.

Completing this step of the results, Figure 3.30 presents the percentage of the chip area occupied by the interconnection architecture for each topology analyzed in this chapter according to the number of cores specified for each SoC, each core being equals 1mm². According to these results, this percentage of chip area is quite small, mainly for the HiCIT architectures. However, usually, in heterogeneous SoCs, the average core size is smaller than 1mm² considered in these experiments (see Table 3.3, Table 3.5 and Table 3.6), since there are some small specific IPs.

Figure 3.30 - Percentage of SoC occupied by the interconnection topologies with the total area divided in wires and cells.



Source: elaborated by the author.

3.5 Considerations

In this chapter one of the proposals of this work was presented. The NoC architecture solution is a hierarchical topology composed of mesh routers on the global level and crossbars on local levels. Many experiments were verified for this solution. When compared to a conventional mesh topology, gains in area, power consumption and latency were obtained. In chapter 5 other analyses related to this proposal will be verified and performance results for HiCIT topology will be presented.

4 ADAPTIVE NETWORKS-ON-CHIP

Modern MPSoCs present a high complexity because they must efficiently handle situations not foreseen at design time. Furthermore, many current NoCs are static, in the sense that their performance and power requirements are defined at design time, for a given application, for example the generation of application specific architectures. As referenced in section 3.1, most state-of-the-art NoC architectures and their design flows provide only a design-time NoC optimization (which topology selection and mapping) for a single application (CHAN, 2008), (PALERMO, 2007), (BJERREGAARD, 2006). However, in the works proposed in (CHOU, 2011), (HU, 2006), (LAI, 2011), (YAN, 2008), (YU, 2010), each architecture can have different requirements, depending on the target application. Therefore, designing a specific NoC to cover all possible changes or updates of an application means always considering the worst case scenario, resulting in an oversized and expensive router. In this case, the designed solution would present excessive power dissipation for the average case (CHEN, 2003). Moreover, designing particular communication architecture for specific applications would mean that many important decisions would have to be taken at design time, hence preventing scalability and online optimizations. Once again the need for adaptive NoCs emerges. In an NoC, different design decisions can be taken, considering a variety of aspects, such as: the number and depth of buffers, router topology, switching mechanism, mapping, arbiter amongst others. Prior works have already shown that for different NoC configurations (topology, buffer size, and bit-width), there is no single NoC configuration capable of providing optimal performance across a large range of applications (MATOS, 2011b). Therefore, the ability of the network to adapt at runtime is mandatory, and examples of such architectures can be found in (FARUQUE, 2008), (MATOS, 2011c), (NICOPOULOS, 2006).

There are different sets of constraints to be considered when is given adaptive capabilities for NoCs. For instance, instead of strictly aiming at speed, designers increasingly need to consider energy consumption. In the next section, different proposals of adaptability

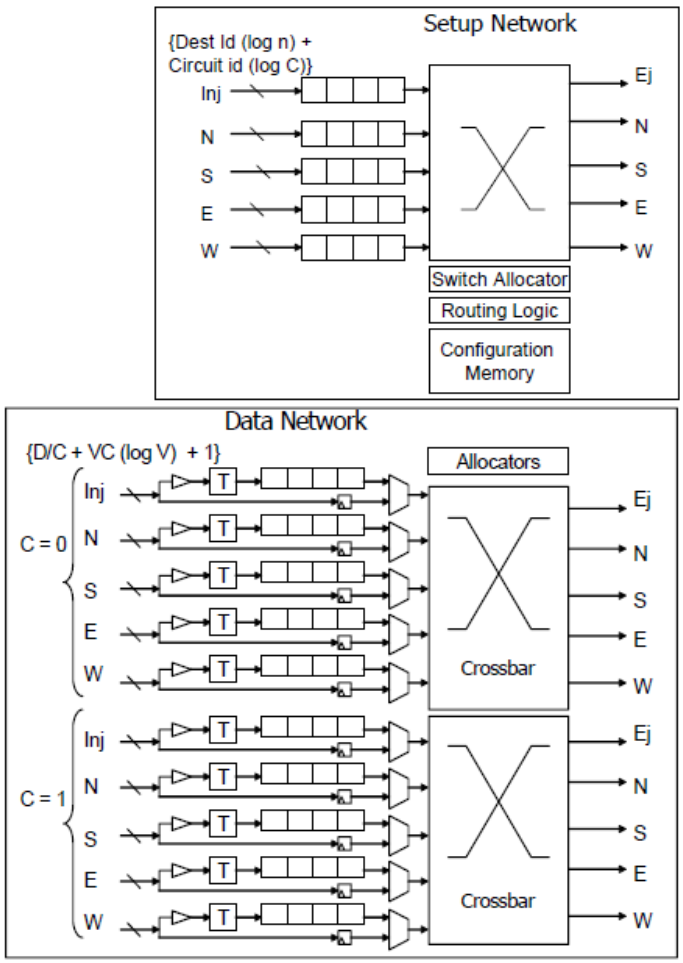
in network-on-chip will be discussed and after an adaptive proposal for NoCs will be presented.

4.1 State of the art adaptive switching NoCs

In order to achieve the performance requirements of different systems, some works propose the use of adaptability in the switching mechanisms, such as VIP (MODARRESSI, 2010), HCS (JERGER, 2008), and EVC (KUMAR, 2008). In all these solutions, a type of circuit switching (CS) is implemented together with the packet switching (PS) mechanism.

In (JERGER, 2008) a router architecture including circuit switched (CS) and packet switched (PS) networks, named HCS (Hybrid Circuit Switching), was proposed. In this work the authors used two separated mesh networks: one for data and another for setup. Furthermore, the data network has a different architecture for each switching possibility (CS or PS), with specific physical channels for each architecture. The setup network works similarly to the traditional circuit switched one, where the paths to send the messages are first constructed by the architecture. Afterwards the path configuration is defined and the flits are sent without the need to wait for an acknowledgment. In that proposal, the area and consequently the power dissipation of the architecture are high, since no resources are reused and the architecture (and hence power) is almost tripled, when compared to a wormhole design. HCS architecture is illustrated in Figure 4.1 with the two networks.

Figure 4.1 - HCS architecture.

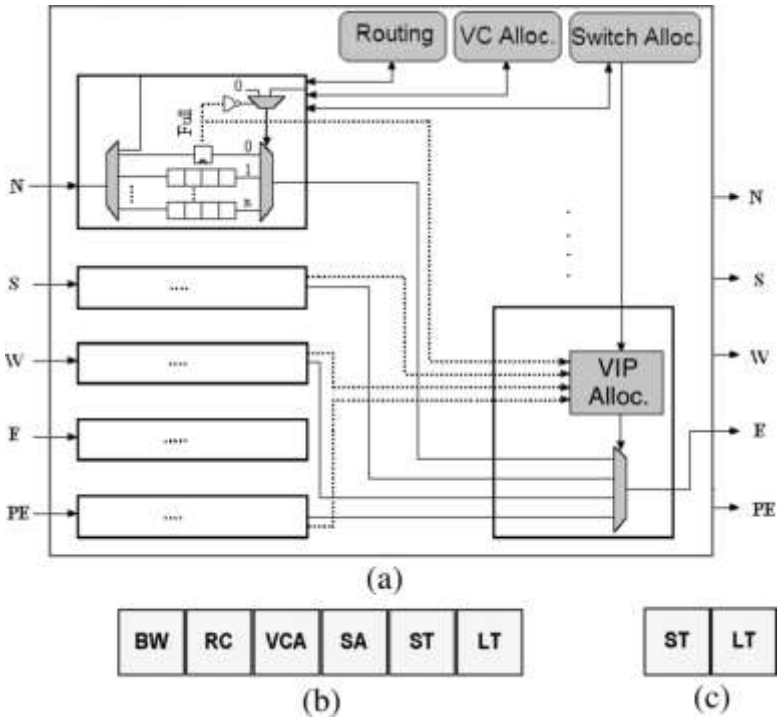


Source: JERGER (2008).

HCS presents an initial setup configuration to construct the path to transmit the messages, similar to traditional circuit-switched networks. Proposals with an initial setup, sometimes do not obtain advantages in terms of latency due to this configuration time. The routers in that proposal have a setup network with three pipeline stages. The setup network does not present the pipeline stage for VC, since the traffic in that network is slow. However in that case, it uses a specific network to reduce the time with the circuit configuration. That strategy considers a configuration packet to create the circuit. In this case, the packet travels along path up to the destination and it does not require a confirmation message. Besides, according to Figure 4.1 the data network is composed of two routers and the use of the first or the second router depends on the indication of the flit as circuit or packet. Similar to the MINoC proposal that will be presented in the next section, HCS deactivates the circuit configuration when it finds another circuit in the path. In this case, the switching is configured to packet and a message is sent to the source in order to alter the method of data sending.

In (MODARRESSI, 2010), the architecture called VIP (virtual point-to-point) has been proposed. In VIP, similarly to the work presented in (JERGER, 2008), the connections between two cores can bypass intermediate routers. The difference of VIP is related to the virtual channels, where the channels are defined as circuit or packet. In that architecture, one channel presents a register with 1-flit buffer, as can be seen in Figure 4.2. Although this design uses VCs, VIP does not allow the sharing of the same link in the circuit-switching mode. In this case, in order to avoid starvation, a threshold between cycles for VIPs and for PS is defined. In this manner, when the number of cycles reaches the threshold and there is a concurrent message for the same output channel, the switching is alternated. The advantages obtained with VIPs are achieved by bypassing the pipeline stages of the PS mode whenever it is in the CS mode, according to the illustration in Figure 4.2(b) and (c). In this manner, the latency and power consumption are reduced. However, this strategy presents some limitations. Firstly, the definition of the VIPs in the dynamic manner requires a centralized control, which is not always possible to implement, especially in large systems. This fact also impacts performance, since it needs to define the VIPs before the application runs. As well as this, a setup network is required to construct the VIPs, such as HCS. The use of a setup network largely increases the area and power consumptions.

Figure 4.2 - (a) VIP architecture, (b) pipeline stages in the packet-switching mode, (c) pipeline stages in the circuit-switching mode.



Source: MODARRESSI (2010).

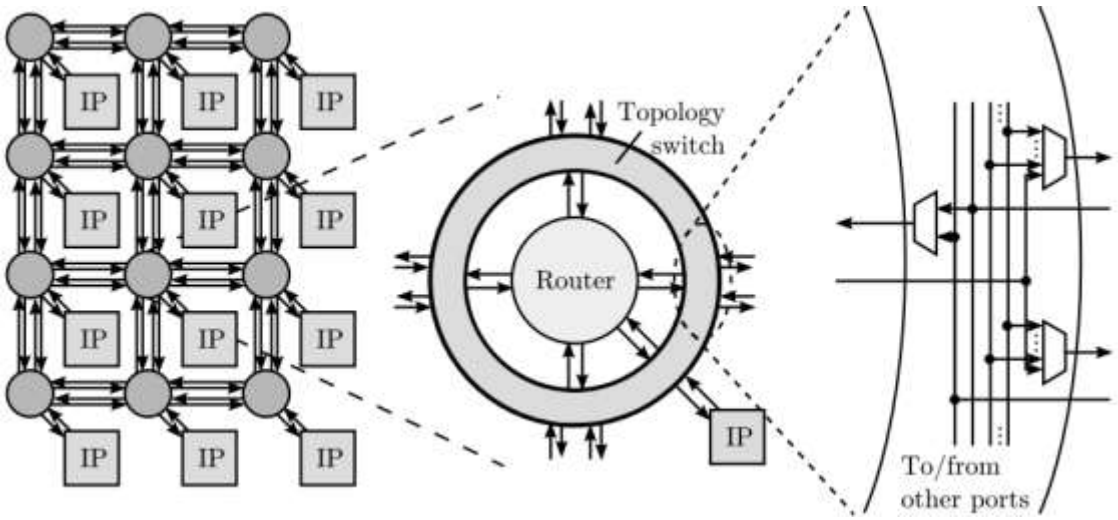
In a previous work (MODARRESSI, 2009), the authors also proposed a hybrid packet circuit switching, where two sub-networks are inserted in the NoC: Pnet and Cnet. The Pnet network is used to implement the traditional packet switching mechanism, while Cnet is dedicated to the bypassing scheme. A setup network, Snet, is used to establish the circuit. That solution uses Spatial Division Multiplexing (SDM) to divide the links. Thus allocating a subset of the links and dividing the modules of the router to each circuit. That work states that the area overhead is small (10%) when compared to the conventional router; however, these results are not presented on paper. Moreover, as the original router is complex as it presents VCs, controls and additional registers, the percentage increment seems to be insignificant.

In (KUMAR, 2008), the authors proposed EVCs (Express Virtual Channel), another solution that allows bypassing intermediate routers along the path. That strategy is similar to (MODARRESSI, 2010): the architecture considers VCs, there is a register specific for the EVCs and the gains are due to the reduction in the pipeline stages when in the circuit mode. However, they also start from a very complex network model to propose the EVCs, imposing significant area and power overheads. The solutions proposed in (JERGER, 2008), (KUMAR, 2008), (MODARRESSI, 2010) imply excessive areas, since in order to obtain the projected advantages, the area is more than duplicated when compared to a simpler and traditional packet switching strategy. These proposals provide latency reduction, since they use many pipeline stages in the router for packet switching (around 5 pipeline stages) and fewer pipeline stages for circuit switching, due to the bypassing of stages. Even so, several of these switching adaptability solutions use virtual channels, requiring a large amount of resources, as previously mentioned. Thus, considering the use of all these resources, the power dissipation required could preclude the usage of these solutions for embedded applications.

In another proposal, ReNoC (Reconfigurable NoC) (STENSGAARD, 2008), the NoC router allowed reconfiguring the NoC topology according to the needs of the application. This proposal uses a set of multiplexers external to the packet switching router, in order to allow the bypassing of intermediate routers and directly connect two cores through long links, thus changing the original topology. However, the restriction of that implementation concerns the long links established when many routers are bypassed, since the long wires degrade the performance, limit the maximum operating frequency and compromise the power consumption. This feature allows the direct connection of PEs with high bandwidth, reducing the latency in this transmission. Figure 4.3 outlines the Topology Switch (TS) module that is integrated in each router. Figure 4.4 shows two examples of logic topology configuration from the TS. Hence, ReNoC allows the assumption of different topologies, according to the

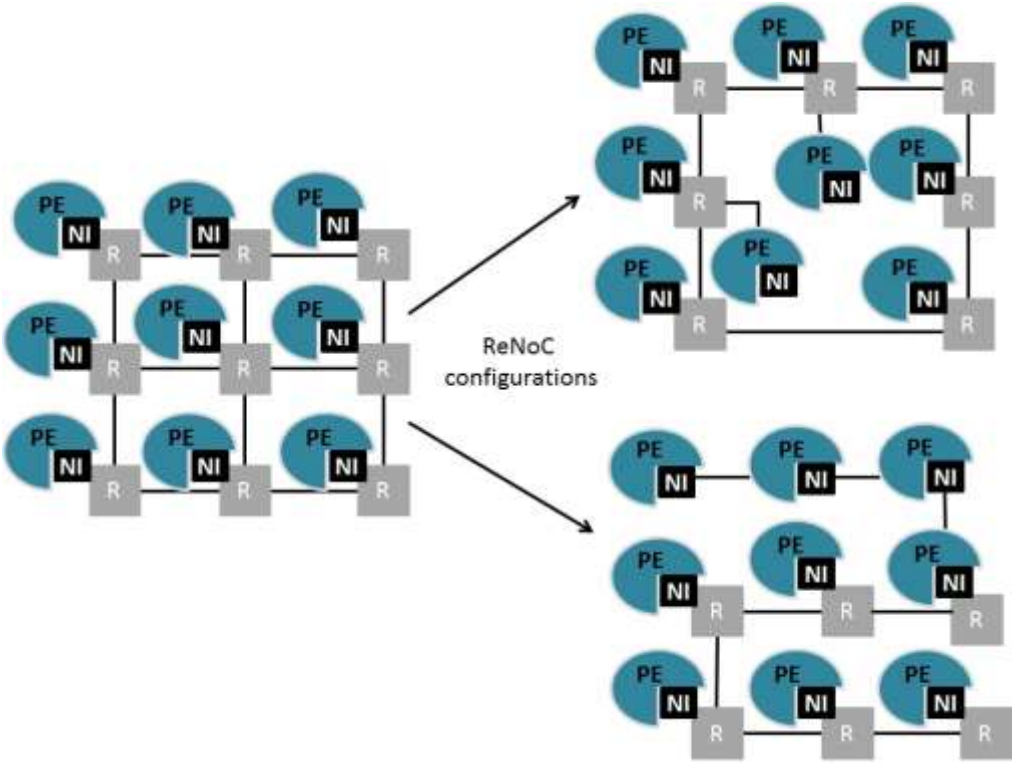
needs of an application, with only an initial configuration time in a completely transparent way for the systems. The strategy in the ReNoC is to make direct connections between the PEs, switching off the routers bypassed and allowing a reduction in power consumption of the interconnection compared to a mesh topology.

Figure 4.3 – ReNoC architecture router with the proposed multiplexers.



Source: STENSGAARD (2008).

Figure 4.4 – Two possible configurations from ReNoC architecture.

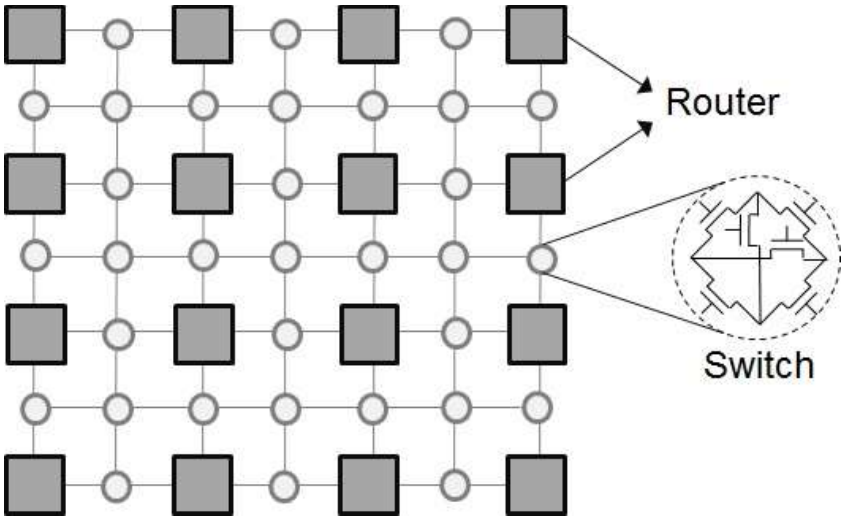


Source: STENSGAARD (2008).

Another similar strategy to ReNoC was presented in (MODARRESSI, 2011). In that work, a topology suited to each input application is appropriately configured. The

reconfiguration of the proposed NoC architecture is achieved by inserting several inter-node switches (switch box). The routers are not directly connected to each other, but are connected through simple switching boxes, as depicted in Figure 4.5. Each square box represents a network node which is composed of a processing element and a router, while each circle represents a configuration switch. Figure 4.5 also shows the internal structure of the configuration switch, consisting of a few simple transistor switches that provide the connection between incoming and outgoing links. That proposal solves the problem of long interconnections experienced by the ReNoC solution, as it uses latches in the architecture to act as registers in the case of long links that could damage the operating frequency. Nevertheless, for that proposal a large number of small switches and latches are required, as well as the conventional routers, considerably increasing the costs of the architecture. This proposal applies a table-based routing scheme, which also has an impact on the power consumption. Another important aspect is related to the reconfiguration of a given topology for other applications. The authors do not discuss this possibility therefore the applications are mapped statically for the architecture. Table 4.1 presents a summary of the techniques related to the switching adaptability in NoCs.

Figure 4.5 – Use of switch box to compose the architecture presented by (MODARRESSI, 2011).



Source: MODARRESSI, 2011).

Table 4.1- Comparison among the adaptive strategies related to the switching.

Author	Fully Dynamic	Strategy	Benefits	Costs
(JERGER, 2008)	No	CS and PS	Latency	Extra NoC
(MODARRESSI, 2009)	No	CS and PS	Latency and Throughput	Extra NoC
(MODARRESSI, 2010)	No	CS and PS	Latency	Complex architecture (use of VCs)
(STEENGAARD, 2008)	No	Topological reconfiguration	Latency	Long interconnection
(MODARRESSI, 2011)	No	Topological reconfiguration	Latency and Throughput	Additional Switches
MINoC	Yes	PS, UCS, BCS	Latency and Throughput	Minimal

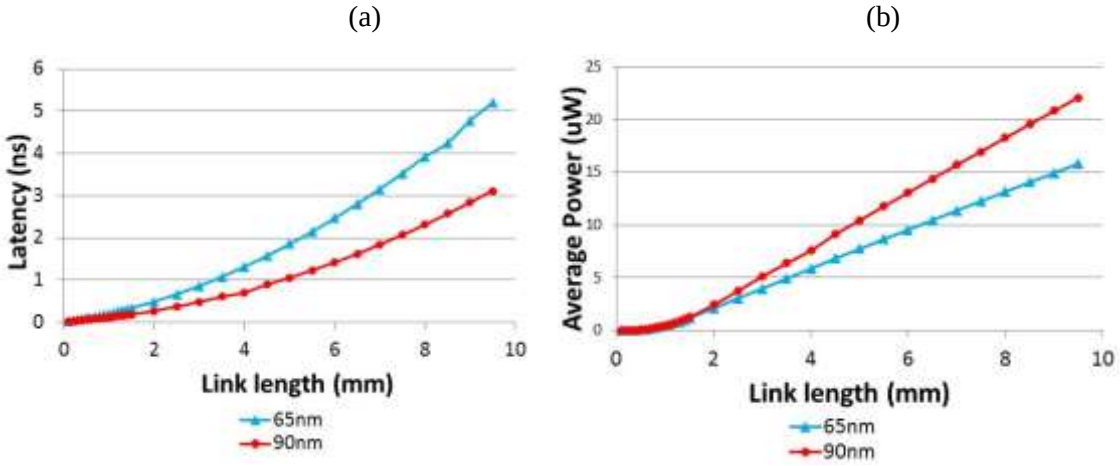
All proposals presented in this section have similar goals and use some type of circuit-switched strategy to improve performance, but many of those proposals use excessive area resources and, in some cases the real costs of the proposal are unclear. Besides this, the solutions presented by (MODARESSI, 2011) and (STEENGAARD 2008) can configure long wires that limit the maximum frequency. The proposal presented in the next section attempts to solve all these problems. Besides, the majority of the solutions do not present an automatic mechanism of adapting the system, which in fact impedes a complete analysis of the strategy's benefits.

4.2 Switching Strategic analysis

It is clear that a better latency result is obtained whenever the cores are directly connected, such as in a point-to-point interconnection. Nevertheless, this solution does not present flexibility when a core needs to communicate with many others. The throughput may be compromised if a pipeline strategy is not used. Another problem is related to the long links that are generated in this kind of interconnection, and this same problem can occur in circuit switching (OGRAS, 2005). In the recent technologies, the long links can present a high degradation of latency and power consumption, (JERGER, 2008). In order to have the scaling benefits of packet switching together with the high speed of circuit switching, while at the same time avoiding the pitfall of high latency degradation in long communication links, this thesis proposes an adaptive switching mechanism at runtime.

Following the analysis reported in sub-section 2.3, the latency and power dissipation according to the wire increase were verified. These studies were done using Hspice for 65nm and 90nm process technology with the parameters according to (PTM, 2013), and the same RLC-π model (SAKURAI, 1983) as used before. Figure 4.6 presents these experiments. The first conclusion that can be drawn is that the latency degrades with the reduction of the technological scale. In the power analysis as the wire length increases, so too does the consumption but on a larger scale. The fact that 65nm technology presents a smaller increment than 90nm process technology is due to the reduction in voltage, since the voltage reduces with the technology scale. For these results, according to the wire model, the resistance is around 450Ω for 65nm and 240Ω for 90nm. The capacitance is 170fF for 65nm and 200fF for 90nm.

Figure 4.6 - (a) Latency and (b) Average power consumption considering the wire length for 65nm and 90nm.



Source: elaborated by the author.

From all these analyses, the impact of the wires on the designs is very clear. Because of this solutions that allow circuit-switching, as presented in the previous section, must be designed taking these factors into account.

In order to avoid long links, this thesis proposes a mechanism called MINoC (Multi-Interconnections Network-on-Chip). MINoC allows three switching configurations: packet switching (PS), buffered circuit switching (BCS) and unbuffered circuit switching (UCS). In order to provide these switching strategies at a low cost, the architecture reuses all resources of a conventional router and includes a minimal control to provide the self-configuration of these switching possibilities. The definition of the switching mode considers the current status of the network and does not use the conventional initial setup, as demonstrated by other architectures.

Similarly to strategies presented in the state of art section, another strategy allowing changing the switching strategies to transmit the messages will be presented in this work. However, the proposed solution presents an innovative strategy able to obtain better performance results thanks to the control of the usage of long wires, caused by the circuit switching strategy.

In the proposed approach, the long wire issue is handled by the insertion of a time barrier in the circuit path. When the desired NoC channel is busy, the transmission is changed to packet mode. On the other hand, if the path is free the UCS is set and the BCS mode is used only when a long wire is detected.

The BCS uses the circuit switching strategy, but in this case one register is introduced in each router in order to restore the signal and maintain a high operating frequency. The latency of each switching strategy in an NoC can be defined according to equations 5, 6 and 7 for PS, BCS and UCS respectively.

$$L_{PS} = (period \times pipe_{PS}) \times hops + L_W + T_C + T_S \quad (5)$$

$$L_{BCS} = (period \times pipe_{BCS}) \times hops + L_{WS} \quad (6)$$

$$L_{UCS} = (3 \times L_{MUX} + L_{AND} + L_W) \times hops \quad (7)$$

where *period* refers to the clock period, *hops* is the number of intermediate routers between the source and destination, *pipe* is related to the pipeline depth (in number of cycles), T_C is the time due to the contention in the network, T_S is the time of setup related to the configuration spend by the header flit (handshake time), L_W is the wire latency, L_{MUX} refers to the latency of a multiplexer, and finally, L_{AND} is the latency of one AND gate.

When considering a specific source and destination, the number of *hops* and setup time will be the same, independent of the switching strategy. The number of pipeline stages differs according to the switching strategy: 5 cycles ($pipe_{PS}$) are needs for the PS, while for the BCS only 2 cycles are required. The contention time (TC) depends on the network traffic conditions. For now, just to compare the methods, an NoC without contention will be considered. The latency related to multiplexers and AND gate of the equation 7 refers to the UCS implementation of this work. In this manner, the latency to transmit one flit in the UCS mode is the same latency of link from source to the destination.

In order to understand the need for each one of these switching strategies, Figure 4.7 shows the latency and throughput for each situation. In this case, for the same packet, different switching mechanisms can be chosen along with the path, as will be detailed in the

next section. The results of Figure 4.7 were obtained considering the time to transmit one flit without any contention along the path. These analyses were found for a 2D-mesh NoC. For the UCS, it considers the delay from Hspice simulations for 65nm process technology. The same aspects described in section 3.3 regarding the wire results from Hspice were used here, and to obtain the link's length, around 1mm² of core area was considered as illustrated in Figure 4.8.

From equations 5, 6 and 7, the latency and throughput results were verified. For these analyses the results were obtained as the number of hops increases. The latency results for PS and BCS were found from the number of clock cycles needed to reach the destination core. The difference between these two possibilities is that BCS does not need to use the handshake protocol, since, when this configuration is defined, the packets are sent directly to the next router using only an intermediate flip-flop (FF). In order to understand the difference of the results according to the clock cycle, two operating frequency were analyzed: 600MHz (Figure 4.7 (a) and Figure 4.7 (b)), and 1GHz (Figure 4.7 (c) and Figure 4.7(d)).

These plots illustrate that for longer interconnections it is better to use the BCS as the UCS delay is larger than the clock period. Besides, UCS presents a limitation concerning the throughput, since it is advantageous to use this switching for only a few hops once no pipeline stage was implemented for this strategy (these values depend on the operating frequency). For 600MHz, UCS mode presents much lower latency than PS and BCS. As the frequency increases (Figure 4.7(c)) the BCS curve is closer to UCS, so, from a number of hops, BCS becomes more advantageous. This occurs because as the clock cycle is smaller, the pipelining modes have the latency decreased. With regard to the throughput, as the frequency increases, more disadvantageous is to use the UCS mode. As the other modes (PS and BCS) have pipeline stages, in a network without contention, the throughput is the same.

Thus, from these analyses, the conclusion is that to obtain a lower latency, the switching is obtained considering the alternated use of UCS and BCS. The use alternated of BCS and UCS is also observed in the throughput analysis, where after a few hops a buffered strategy is needed in order to obtain a new flit for each cycle time. This possibility was called as ideal switching mode, since it obtains better results for latency and throughput for all cases when compared with the other presented strategies. For these experiments, the ideal latency takes into account one BCS to each two hops in UCS mode.

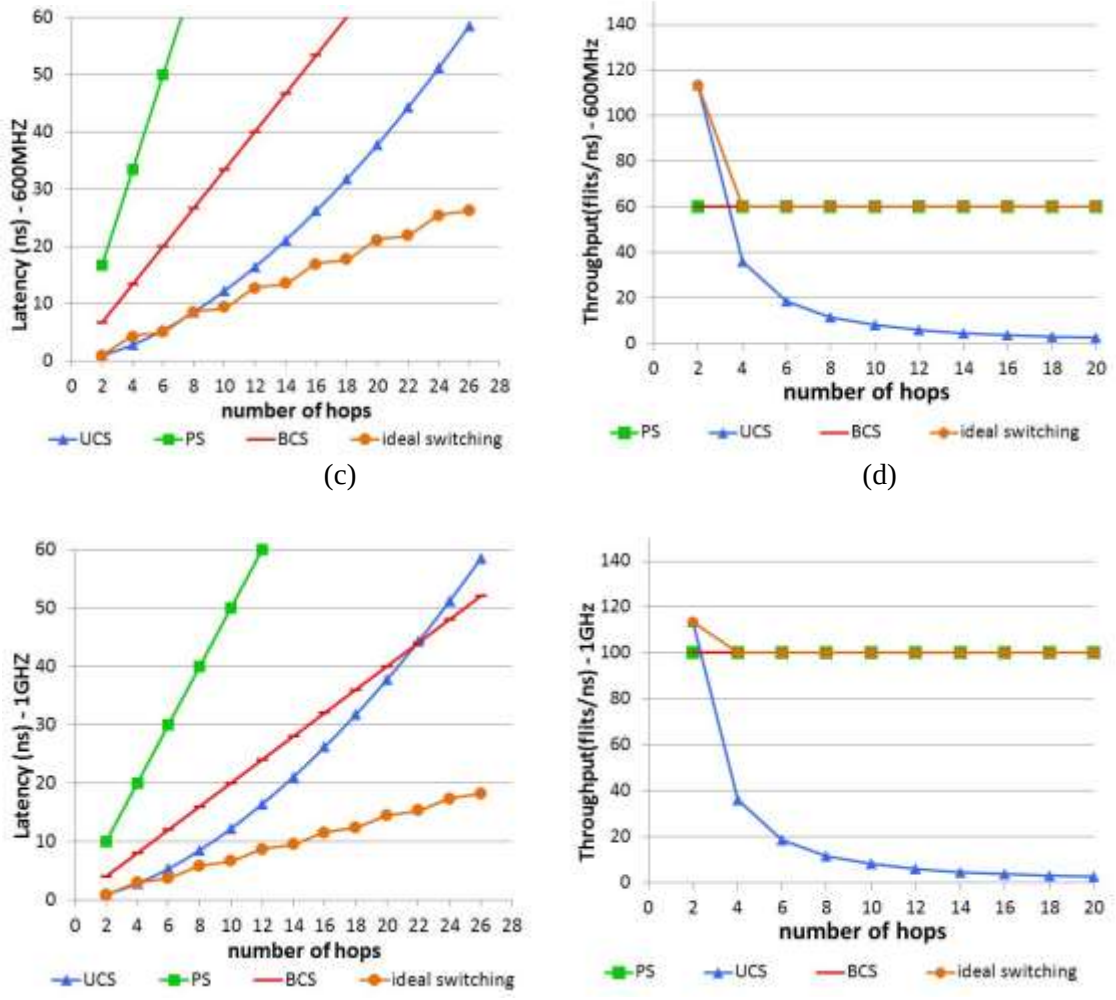
The ideal switching shown in Figure 4.7 is the proposal of this work, where the best results in terms of latency and throughput are achieved using a mixed switching strategy. In

this case, when the UCS reaches the maximum wire length according to the operating frequency, the BCS is set. After this, UCS can be set again for the next routers.

In short, the highest communication rate (UCS) is defined for short paths allowing elevating operating frequencies. Whenever the cumulative wire length is considered long, the BCS is set. However, if part of the required path to transmit the flits is already being used by another packet, the PS is defined. In order to incorporate all these resources effectively, the architecture sums some of the complexities demanding additional hardware, and consequently increasing the area and power consumption. With the aim of reducing the impact of these additional resources, the control mechanism embedded in the architecture is simplified and reuses many components of the original router, instead of using another router as presented in the other architectures. However, in order to lessen these costs, in the next chapter a more intelligent manner of combining this strategy will be presented.

Figure 4.7 - Latency and throughput results considering different switching possibilities for 600MHz (a), (b) and 1GHz (c), (d).

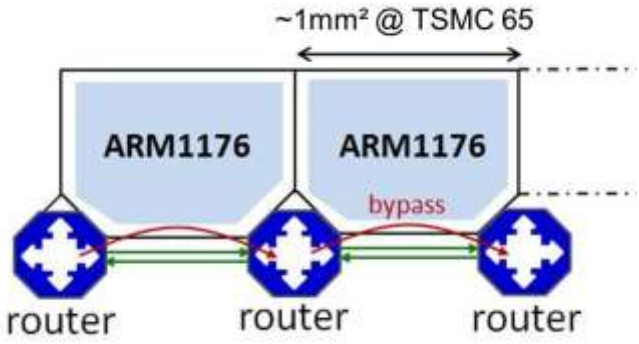
(a) (b)



(c) (d)

Source: elaborated by the author.

Figure 4.8 - Wire length adopted for the experiments for circuit switching mode.



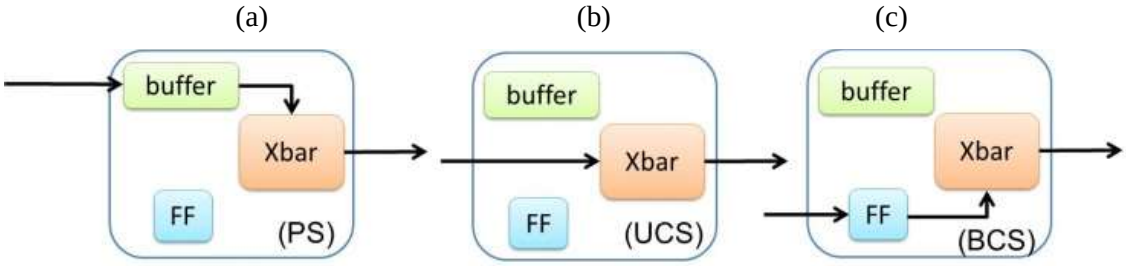
Source: elaborated by the author.

4.3 MINoC architecture: an adaptive NoC

In this section the hardware architecture of the proposed adaptive NoC will be detailed. MINoC (Multi-Interconnections Network-on-Chip) allows three switching possibilities dynamically reconfigurable to avoid long interconnections, while at the same time trying to obtain the optimum performance of the system.

The three switching possibilities are defined according to the NoC traffic. Figure 4.9 shows the three configuration modes that can be defined in each router. The difference between the modes is related to the storage of the flits. In the PS, the flits pass by the conventional buffer (FIFO). If the selected mode is UCS, the flits are not stored in the router and pass directly by the crossbar. In the BCS, a single register (Flip-Flop (FF)) is enough since all flits will follow without interruption to the destination.

Figure 4.9 - Three switching configuration possibilities: (a) packet switching, (b) unbuffered circuit switching and (c) buffered circuit switching.



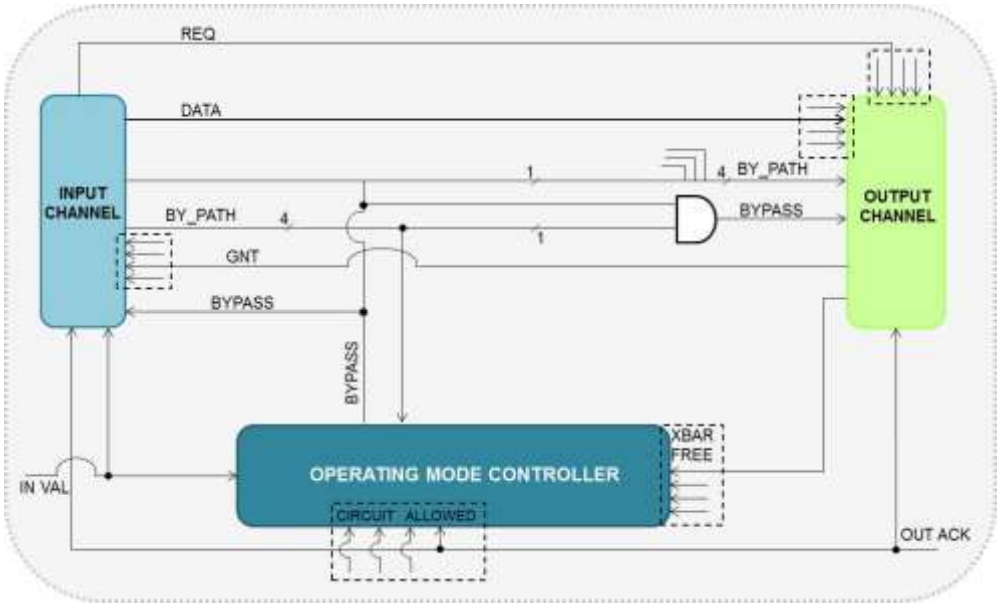
Source: elaborated by the author.

The MINoC router architecture is illustrated in Figure 4.10. Each router port is composed of Input Channel, Output Channel and Operation Mode Controller. In the input control, there is a multiplexer to select if the data will be sent by packet, unbuffered or buffered circuit. It also has an implemented control called *Long Link Controller* (LLC). Each

router port uses this control to identify the wire length between two routers according to the circuit floorplan. This mechanism uses the knowledge of the wire length in the interconnection from the last router to the current router plus the information of the accumulated wire length while in circuit mode. The accumulated wire length is stored in a specific field of the packet header, generating a total wire length estimation. Then, LLC control checks if the estimated delay for this total wire length is higher than the clock period, indicating the necessity to change the UCS mode to BCS mechanism. This process is totally automatic and can be deployed at runtime as will be presented later.

In order to cope with circuit and packet switching strategy, two control signals are used: *bypass* and *by_path*. The bypass isolates the control signals from crossbar, activates the *out_val* and selects the output data multiplexer to circuit switching. The bypath selects the input port with circuit switched enabled to output data.

Figure 4.10 - MINoC router architecture.



Source: elaborated by the author.

4.3.1 Operating Mode Controller

Operating Mode Controller is responsible for defining the packet switching and circuit switching modes, and managing the bypass enable signal of each input port. *Operation Mode Controller* architecture is detailed in Figure 4.11 and takes into account three issues:

- If the input channel has flits to receive (*in_val*);
- If the selected arbiter is free (*xbar_free*);
- If the input port of the destination router can receive data (*circuit_allowed*).

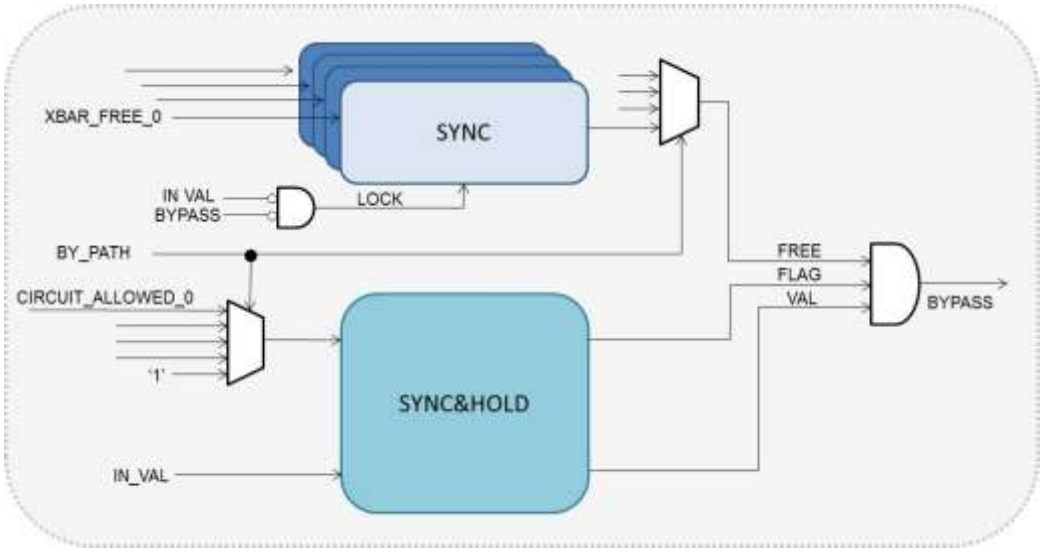
Moreover, a support signal is needed (bypath) to indicate the required output port. Therefore, if the conditions above are met the circuit switching mode can be enabled.

However, there are more considerations in the CS enable management. The first consideration is related to the fact that the router needs one cycle to define the output port destination, which makes it impossible to know whether the crossbar is free or not. To overcome this, the proposed router assumes that all output ports are initially free. In this case, the incoming data through the circuit path is accepted, applying a speculative strategy. If this situation is true, the bypass continues set. Otherwise, the bypass is deactivated, and the multiplexer from input port receives the data. The first flit is not lost because it was previously stored in a buffer. There is no latency penalty in a prediction mistake, because it uses an internal buffer to recover the data in time to use in a packet switching flow.

The modules that assume the speculation are the Sync&Hold and Sync. These modules act according to the *circuit_allowed* and the *xbar_free* signals, respectively (as shown in Figure 4.11). The first module is responsible for holding in one cycle the *circuit_allowed* in '1' in order to delay the information until the correct *circuit_allowed* is known. The second module is responsible for passing the last valid information of *xbar_free*, locking this information after the circuit is set, since the output port is not free anymore.

Another consideration is related to deactivation of bypass, which needs to delay the bypass deactivation information for the output port. This has to be done in order to guarantee that the last flit sent will not be lost. So, the element responsible for holding this information is the Sync&Hold, taking into account the *in_val* information. In addition the crossbar is not deactivated, receiving the requests from other input ports but not serving them until the end of the circuit transmission. The arbiter implemented in this architecture was the Round Robin (RR) that together with the Operating Mode Controller, guarantees a starvation free network.

Figure 4.11 - Operation Mode Controller Architecture.

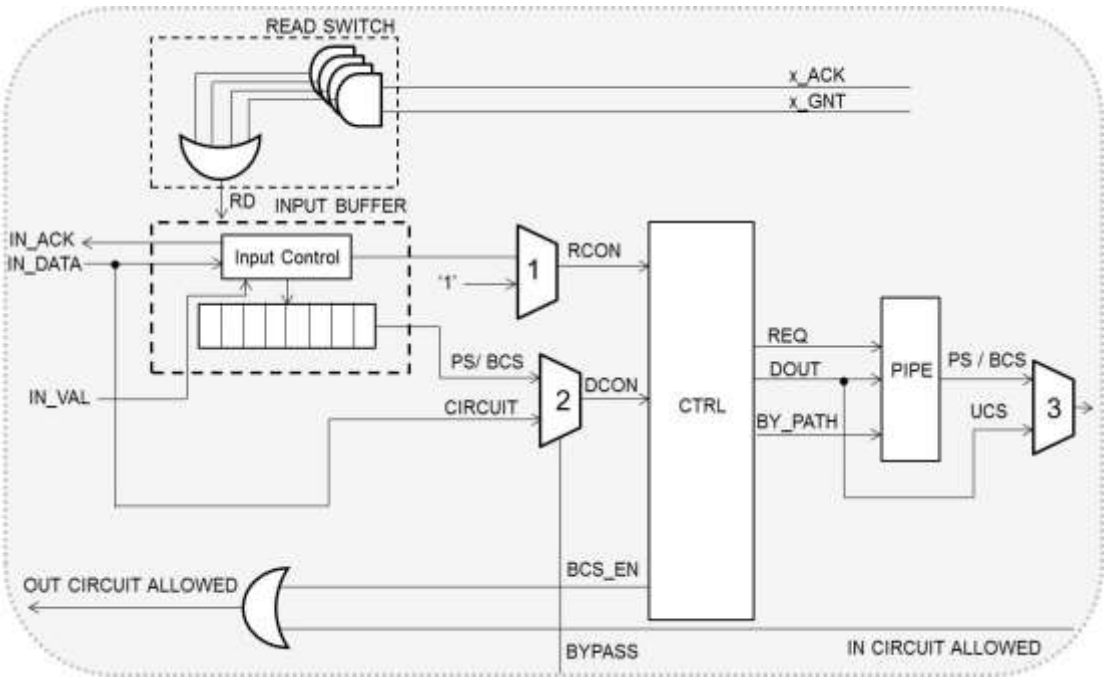


Source: elaborated by the author.

4.3.2 MINoC router input channel

Figure 4.12 presents the input channel architecture. The input channel underwent some modification from the original router architecture to cope with the adaptability strategy. Some multiplexers (1, 2 and 3 in Figure 4.12) were added in the architecture to allow the three switching modes.

Figure 4.12 - Input Channel architecture.



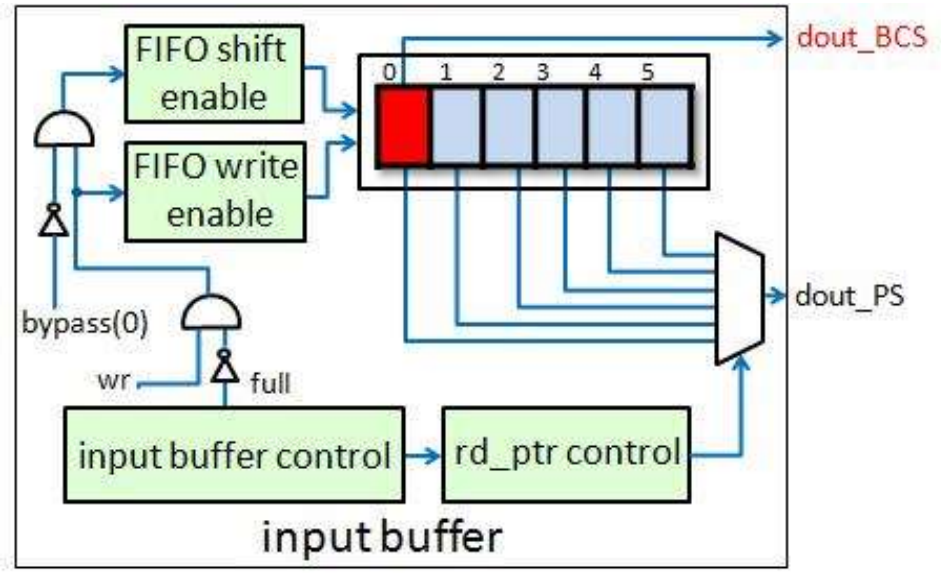
Source: elaborated by the author.

The input buffer was altered in the manner of how the data is stored. The input buffer always writes in the first position of FIFO. This is needed when the circuit mode is turned off and the last flit sent should be stored in the PS mode. In this case, the write pointer is fixed in the same position. When a new flit needs to be stored, the data are shifted. In the BCS mode, as the path is available, a single buffer unit is required. Hence, the input buffer has a control to disable the data shift for this operation mode, preventing the transmission of incorrect data. Figure 4.13 depicts the input buffer architecture.

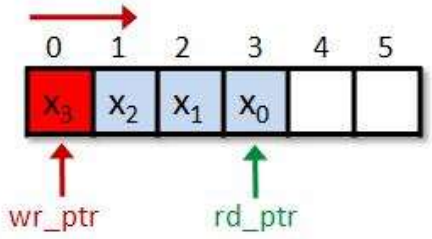
A power consumption comparison between a shift buffer and a conventional circular buffer was done. In this analysis, it was considered 16-bit link width, buffer depth equals to 4 slots for 65nm process technology. For these experiments, considering a power default toggle rate of the tool, the shift buffer did not present a power consumption increment when compared to the original solution (circular FIFO buffer).

Figure 4.13 - (a) Input buffer architecture and the (b) buffer and its pointers.

(a)



(b)



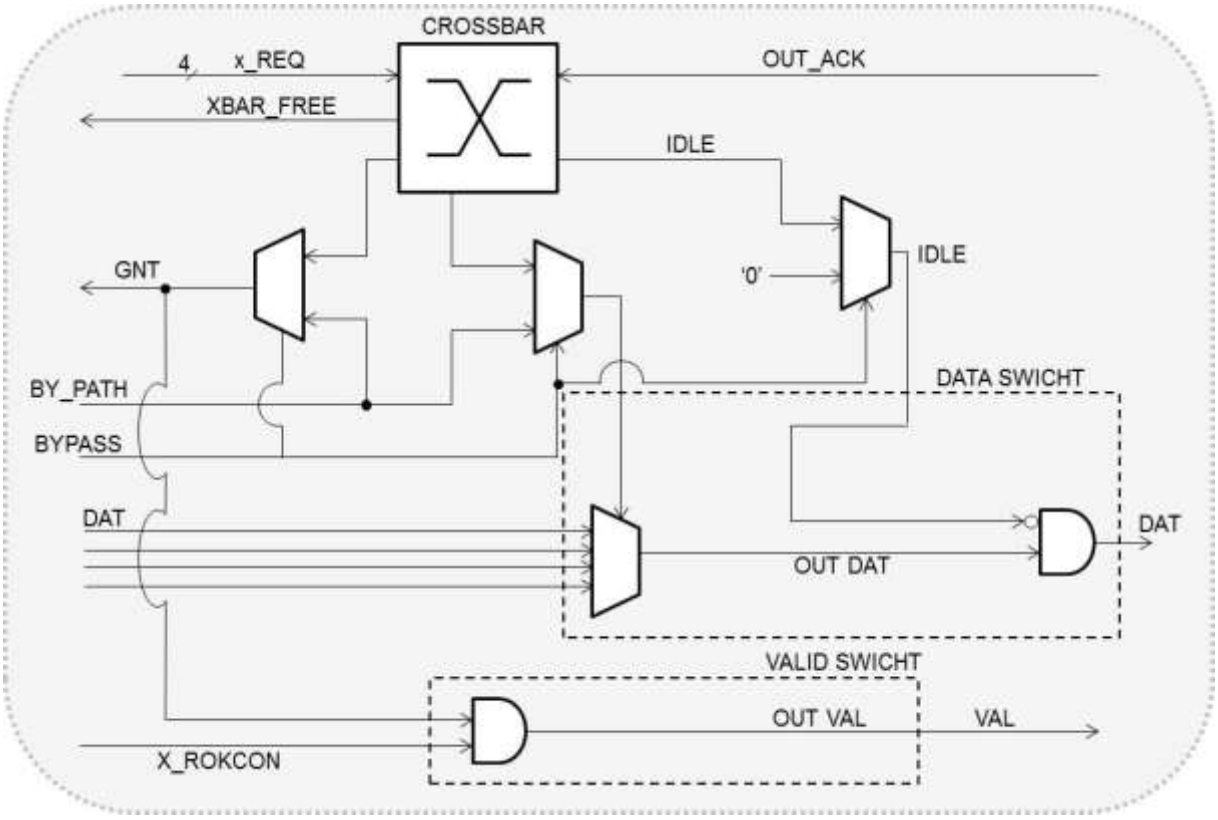
Source: elaborated by the author.

The input channel also presents a control enabling the storage of flits in the input buffer when in the CS modes. This control verifies if the flits in CS need to be stored by verifying the wire length. The total wire length is stored in the header flit of each packet. This wire length information between neighboring routers is obtained previously at design time according to the floorplanning. These values are converted in weighted values and are informed as input parameters to each router. Afterwards, this value is incremented and rewritten in the header field by every bypassed router. For example, if the maximum wire length permitted without violation in the design timing is equal to 3.5 millimeters, then one increment every 0.5 millimeters of wire length and 3 bits to inform this value in the header can be defined. Whenever the accumulated wire length in the header is equal to 111, the circuit switching should be changed to buffered. The conversions of wire length are approximated and what is important is to ensure that the total wire length in the UCS mode always has a smaller delay than the clock period time.

4.3.3 MINoC router output channel

In the output channel architecture, some multiplexers and controls were also integrated in order to disable the protocol used in the output when in the CS modes. In the circuit switching modes the data is directly passed to the output by the multiplexers. As can be observed in Figure 4.14, this architecture has two important signals: bypass and *by_path*. Bypass signal is used to disable the crossbar control during the UCS and BCS modes. Besides, this signal actives the output handshake protocol from the *out_val* signal and selects the data output for the circuit flow. *By_path* signal defines which input channel will have the data flow by circuit switching.

Figure 4.14 - Output Channel architecture.



Source: elaborated by the author.

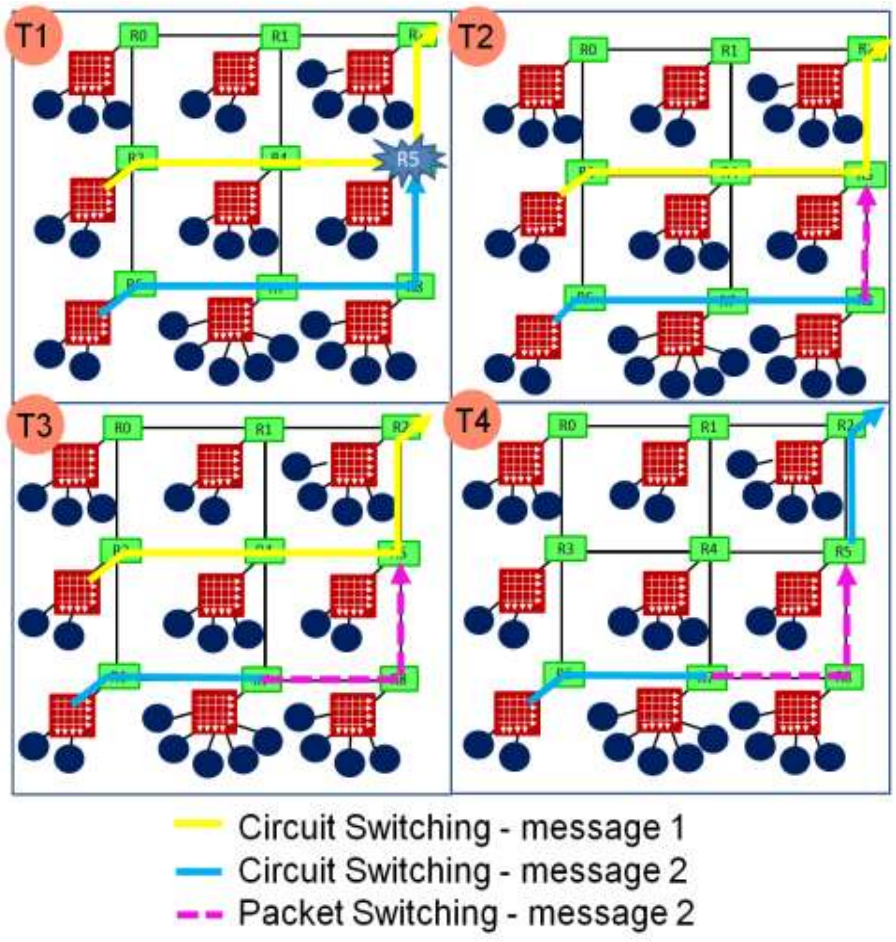
4.3.4 MINoC configurations

In the MINoC architecture, each router is independent from each other, allowing different switching strategies along the path for the same message. When a router changes to the PS mode because another packet is using part of the path, it does not mean that all previous routers defined for this message will work in this same mode, since the buffer can, for a while, guarantee the reception of flits. Only when the buffer is full is this information

sent to the previous router, which also changes to PS mode and so on. In this manner, the bypass is deactivated slowly giving time for the network to release the path and restore the bypass.

The message becomes purely packet switching only if the information of full buffer is propagated until the first node. Figure 4.15 shows an example of MINoC operating. In this case, frame T1 presents two packets in circuit switching mode requiring access to the same router (R5). Considering that message 1 is chosen by the arbiter, the flits of message 2 need to be stored in the input buffer of R5. In this case, the packet switching mode is set in this router (frame T2). When all router buffers are full, the PS is propagated for the remaining routers (frame T3). When the path to the destination node is free, the message is sent in the UCS mode again (frame T4). Thus, whenever there are two packets competing for the same output port in a router, the first packet to request the output channel (or the input port selected by the arbiter when the requests occur at the same time) is switched to circuit and the second packet enters in the packet switching mode.

Figure 4.15 - Example of MINoC configurations.



Source: elaborated by the author.

The advantage of the proposed adaptability is due to the reduction in the number of pipelines in each mode. When the router is operating in PS, 5 cycles per router are required. This number decreases to 2 cycles in the CS modes for the header flit. A time equals 2 cycles is required to initialize the circuit switching in each router: one cycle to send the configuration message and the other to receive the confirmation. For the data flits in the UCS mode, as it presents the control of long links, it is possible to bypass many routers taking only one cycle. The strategy presented in the MINoC architecture differs from the others because it presents an automatic mechanism of adapting the switching dynamically, which it is not allowed in other architectures, and also due to the possibility of using two circuit switching modes.

4.4 MINoC Results

4.4.1 MINoC Performance results

The performance results were obtained with a cycle-accurate traffic simulator described in Java. This is an in house tool based on object orientation interface paradigm, through which all architectural components can be configured to match a diversity of network configurations. The main components are interfaces for: arbitration, control flow, buffering and routing algorithms.

Simulations for this and for the next chapter were performed for NoC with 16-bit link, 4-flit deep buffers, for a total of 5000 packets/node and considering the operating frequency equals 1GHz for all experiments. The definition of the operating frequency value and buffer depth was commented in chapter 3. The data width defined for the links do not impact in the results and so, 16-bit link is a reasonable value when compared to the data width of memories and processors. To verify the network's behavior when the traffic increases, different flit sizes were used: 8, 16, 32, 64 and 128 flits per packet. Analyses of long wire interconnections to define the circuit switching mode were considered. In this case, on average when in the UCS mode, the BCS is set afterwards to bypass two hops in the UCS. According to some studies, one verifies that more than 2 routers can be bypassed running at 1GHz, since a link length with around 4mm can be crossed when operating at this frequency (see Figure 2.14). However, considering UCS every two hops, good results are already achieved. The mappings for the benchmarks are the same defined by ASHiNOC tool in chapter 3 for mesh topologies.

The mesh NoC used in the comparisons is similar to SoCIN network (ZEFEINO, 2003). It considers a XY routing algorithm and a wormhole switching. The bufferization is defined in the input channel and it does not use virtual channels. Some modifications were

implemented in the original architecture, as the increment in the pipeline stages in order to achieve high operating frequencies (1GHz, for example). The input buffer considers shift buffers, as presented in subchapter 4.3.2

The injection rate of packets in the NoC was determined from the operating frequency, link width and bandwidth of each communication. From Equations in (8) the interval in number of cycles to send packets was defined for each communication, where *freq* represents the operating frequency and rate, the core communication bandwidth. The terms in parentheses refers to the units of the equation.

$$freq = cycles / s$$

$$cycles = freq \times s$$

$$t(s) = \frac{link\_width}{rate \times 8} \frac{(bits)}{(bits/s)} = \frac{link\_width}{rate \times 8}(s) \qquad (8)$$

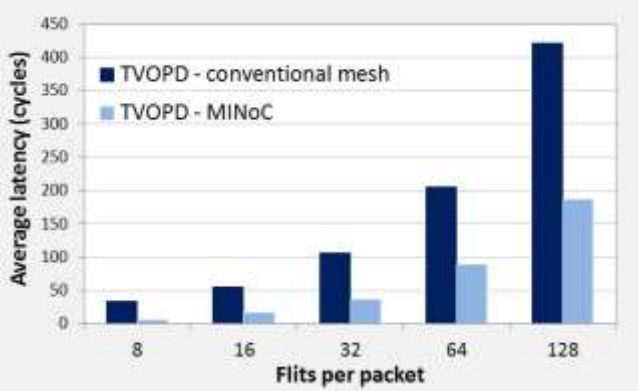
$$cycles = freq \frac{(cycles)}{(s)} \times \frac{link\_width}{rate \times 8}(s)$$

$$cycles = freq \times \frac{link\_width}{rate \times 8}(cycles)$$

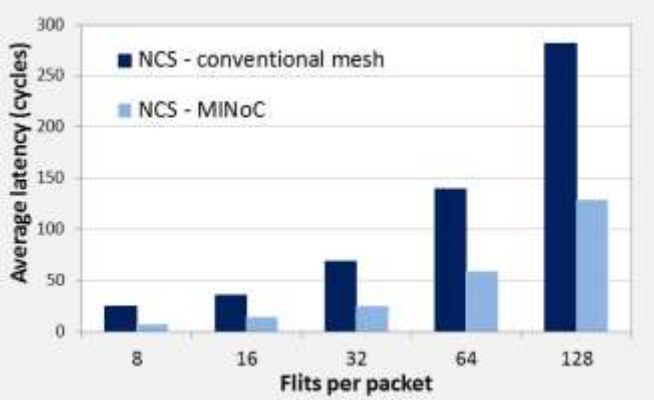
Latency results are plotted in Figure 4.16. As it is proven by these experiments, the MINoC architecture obtains a large reduction in the latency in comparison with a conventional packet switching mechanism. According to these results, the reduction for these three benchmarks is larger than 50% for packets with 128 flits and around 70% for small packets composed of 8 flits. It is possible to conclude that this strategy presents better results for small packets. This happens because when the circuit switching mode is set, if the packet has many flits, other packets using the same path need to wait for longer to receive the output ports. However, if contentions in the network are minimal, more circuits will be formed, and large packets can bring advantages.

Figure 4.16 - Average latency results for MINoC in comparison with a conventional packet switching router for (a) TVOPD, (b) NCS and (c) TMPEG4 benchmarks.

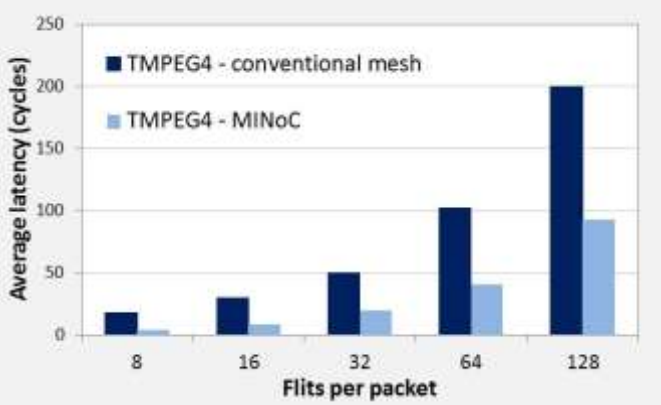
(a)



(b)



(c)



Source: elaborated by the author.

The considerable gains were possible due to the reduction in the number of cycles of the circuit switching mode when compared to the conventional architecture and the possibility of bypassing many routers taking only 1 cycle (UCS mode). Besides, the other great advantage of this solution with regard to the others (JERGER, 2008), (STEENGAARD, 2008), (MODARRESSI, 2011) is the automatic mechanism of adapting the systems, since for

each new packet the architecture detects the possibility of sending the packet in the circuit switching mode and makes this transmission dynamically, according to the traffic on the message path and the floorplan information.

4.4.2 Synthesis Results

Synthesis results for 65nm of process technology were analyzed for the complete system of NCS and TVOPD benchmarks. In order to obtain accurate link lengths, the core areas were considered as black boxes in the synthesis according to the areas informed in Table 3.3 and Table 3.5. In this case, the correct wire information and architectural costs were considered in the logical synthesis (obtained with the RTL Compiler tool) from the parasitic extraction obtained in the physical synthesis (from First Encounter tool). The operating frequency considered in the synthesis was equal to 1GHz as considered in the performance results too. The NoC configuration is also the same previously defined: the buffer depth is equal to 4 and 16-bit link wide.

Power consumption and area results are presented in Table 4.1. The addition of the resources in the MINoC architecture, despite only totaling 1.13% of area increment, in the power consumption this reflects a significant increment. The results of these comparisons were redone and the power consumption increment has been maintained. In order to understand the reasons for this increase, some analyses were evaluated. This increment in the power consumption is not related to the modifications in the input buffers, since the same buffer architecture were considered for both architectures (MINoC and mesh NoC) in these results. Although in the power consumption measured in this work, the shift buffer was not presented an increment in power consumption when compared to the circular buffer. The addition of components in the MINoC proposal represents mostly combinational alterations. This increment largely impacts in the switching activity, since all possible paths are active in the syntheses. In the dynamic power of gates was observed an increment of 35% related to the addition of more inputs in the circuit, that are switched even when a single path is in use (due to MINoC switching possibilities). In the dynamic power related to wires, an increment of 20% was observed related to the addition of wires in the MINoC architecture. Leakage power is increased in the same proportion of area, as expected. In order to allow improvement in performance, MINoC have included many combination circuitries, which incur in redundant switching, once the total switching activity is a relation of the number of input switchings (MENON, 2004). Switching activity for combination circuits is difficult to handle. One

alternative to deal with this problem is called Operand Isolation. Operand Isolation is a technique for minimizing the power overhead incurred by redundant operations (CORREALE, 1995). Some different operand isolation solutions have been proposed in the literature (MUNCH, 2000), (CORREALE, 1995), (BANERJEE, 2006), allowing to reduce up to 40% of the power consumption related to combinational circuitry, with a minimal area overhead (5%). These factors could be the reasons for the high overhead in the power consumption imposed by MINoC. However, the use of techniques as operand isolation is not in the scope of this work. As a result of this, and together with all the studies carried out in this work, another solution was designed and will be presented in the next chapter.

Table 4.2 - Synthesis results for MINoC proposal compared to a conventional router for NCS and TVOPD benchmarks.

Benchmark	**mesh NoC**		**MINoC**		**Overhead (%)**	
	Area (um^2)	**Power (mW)**	**Area (um^2)**	**Power (mW)**	**Area**	**Power**
NCS	563587	61.48	569931	101.72	1.12%	65.45%
TVOPD	1619684	184.85	1637931	288.62	1.13%	56.14%

4.5 Considerations

In this chapter the importance of adaptive solutions in MPSoCs was presented and a dynamic solution for NoCs was proposed. It allows adapting the switching of the routers according to the NoC traffic conditions. The most relevant consideration of this strategy is that the switching mode selection considers floorplan information. In this case, the appropriate selection of each mode takes into account the long links. The adaptability supports three dynamic switching modes. Two of these modes were defined for the circuit switching (buffered or unbuffered). This solution obtains large improvements in performance, reducing the latency to around 70% for small packets when compared to conventional NoC architecture. To provide these performance advantages, the architecture was modified, mainly by adding more controls. Because of this, MINoC consumes a lot of power. In order to combine the advantages of this architecture with the advantages of HiCIT, another strategy will be proposed in the next chapter.

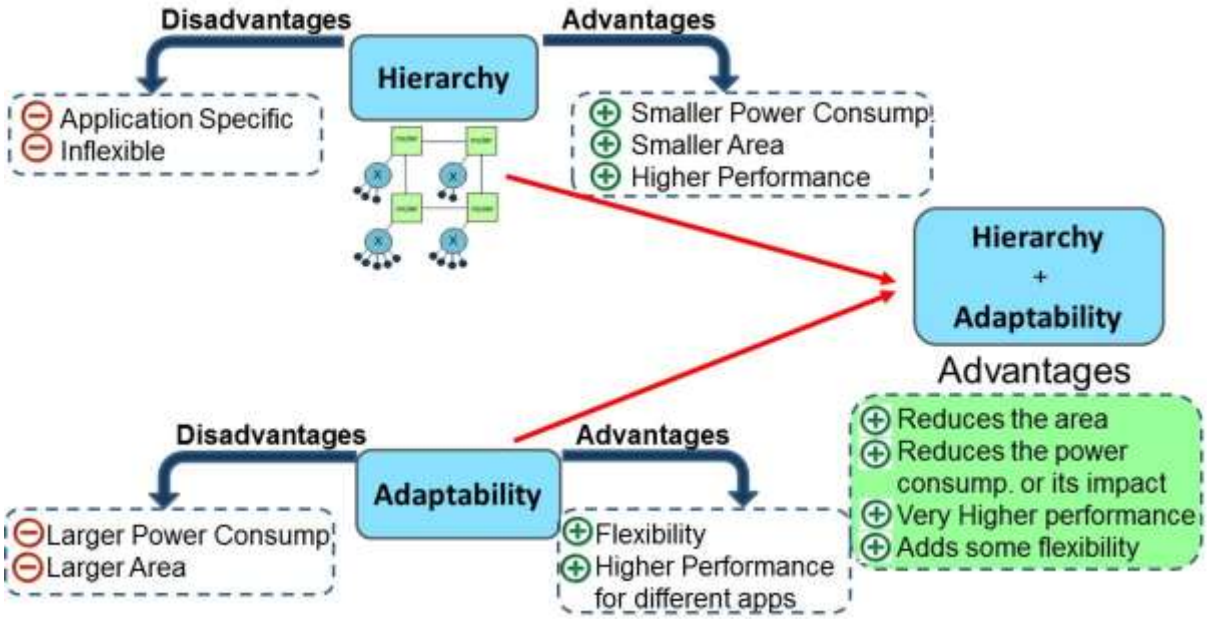
5 HASIN – HIERARCHICAL AND ADAPTIVE NOC

According to the proposals presented in chapters 3 and 4, it could be observed that each one of them presents advantages and disadvantages in different aspects. In a system composed of a large quantity of cores, some strategy to reduce the network contention and the number of hops to transmit the messages is required. As commented in chapter 3, one interesting solution is the use of hierarchy. However, the primary problem of this decision is to design a hierarchical topology for a specific application and later, this application requires updates or changes in the communication rates, affecting the system performance. There are many scenarios of MPSoCs, such as:

- systems designed with a fixed architecture for a specific application;
- systems where the cores can receive remapping of tasks;
- systems designed to run different applications;
- systems that can be updated or present changes in the communication rates.

The proposal presented in this chapter covers mainly the last case: systems that can be updated from an initial design. These situations are very common, for example, one can imagine an application designed for a specific architecture, however, some cores can receive more functions which were not foreseen at design time (the system can be updated and processors can receive more tasks, for example) and then, the interconnection device will need to be fit with this new situation. Thus, it is possible to achieve this interconnection solution integrating the two proposals of hierarchy and adaptability, such as depicted in Figure 5.1.

Figure 5.1 - Ideal NoC proposal is obtained with hierarchy and adaptability.



Source: elaborated by the author.

The use of the hierarchical solution like HiCIT with the MINoC strategy was called HASIN - Hierarchical Adaptive Switching Interconnection Network. With this solution, it is possible to mitigate the impact in the performance results whenever the system suffers any changes in its communication behavior. In this manner, the use of a specific topology reaches the best solution in terms of performance and power consumption from a mapping for an initial core communication graph. Taking onto account that some cores of an application can have their rates increased due to an update. If the increase in the bandwidth occurs in the communications between cores inside of the same cluster, as the crossbar already gets the maximum communication rate, it will have minimal or no degradation in the performance. However, if the messages increase in the global level, the adaptive strategy can reduce the loss of performance. This strategy foresees situations where the application behavior is not radically changed, but, it allows supporting different functionalities of an application.

Thus, thanks to its hierarchical approach, HASIN architecture can cope with specific communication behaviors. However, it also presents flexible features to support different traffic patterns that can be combined with a suitable mapping and routing algorithm. The topology of this strategy is the HiCIT and the MINoC adaptability is implemented in the global level, in the mesh routers. In this case, each class of applications needs to have an appropriate mapping and adequate crossbar granularity. Such solution gets reduction in the NoC power dissipation and the system performance is improved, since when the cores are in the same cluster, the latency to transfer the packets for many routers is removed, and, in this

case, a simple switch protocol is used. At the top level, the improvement in performance is sustained by the switching adaptability. In the next section, the advantages of this solution compared with other proposals will be presented.

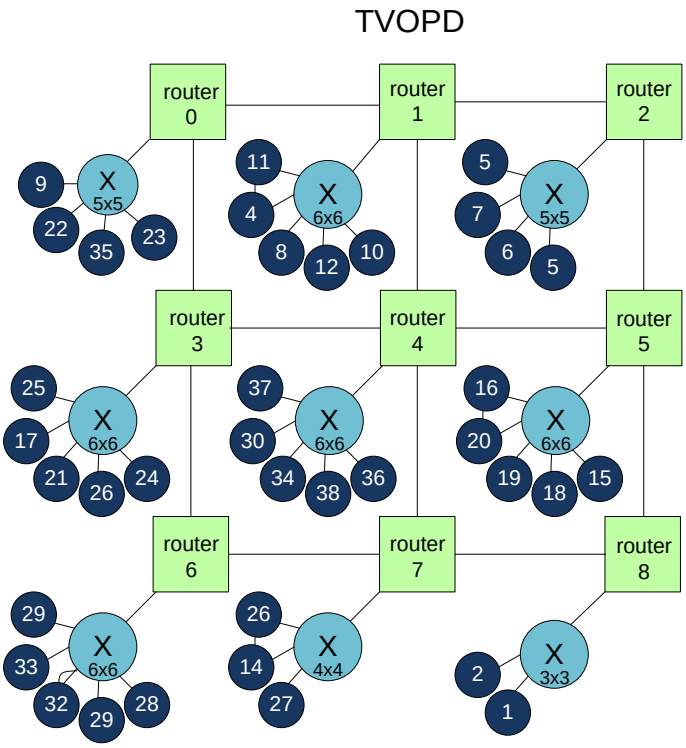
5.1 Experimental Results

5.1.1 HASIN Performance Results

The performance results were obtained with the same tool used in chapter 4. The same configurations defined in chapter 4 were also used in this chapter, such as operating frequency, NoC parameters, injection rate and others.

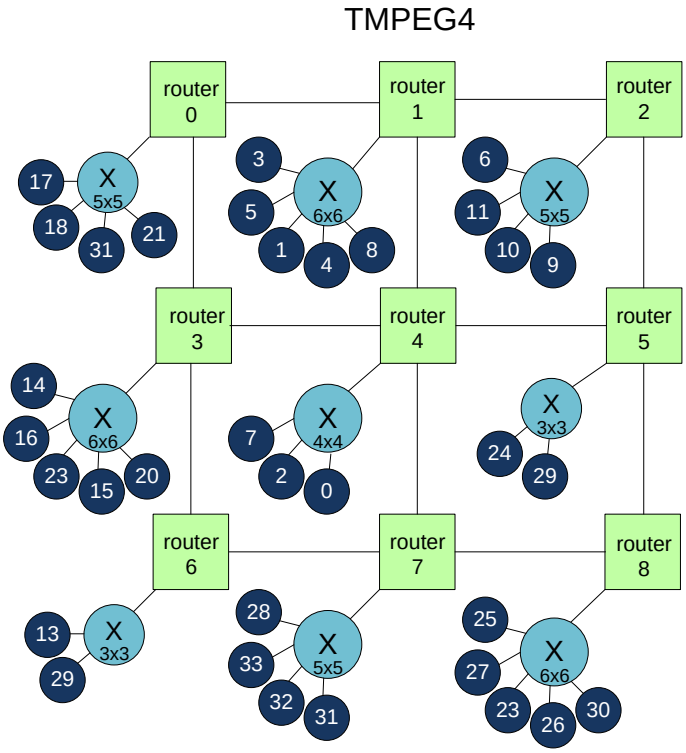
For these experiments, the same benchmarks considered in section 3.4 have been analyzed (TVOPD, NCS, TMPEG4). However, from the mappings for these benchmarks presented in Chapter 3, some experiments proved that there were a lot of communications at the HiCIT global level. As the use of the hierarchical topology is just to concentrate the communication cores with high bandwidth in the same cluster, a manual mapping according to the core bandwidths were verified for the experiments of this chapter. These mappings were defined only for HiCIT topology, considering the tool to map the applications for the mesh topology. For these new mappings, only small clusters were defined (maximum of 8 cores per cluster). As most of cores of these benchmarks are small, according to Figure 2.11, a high operating frequency will be able to be ensured. The mesh mappings were maintained as presented in section 3, since for this topology, communicating cores of the benchmarks are close according to the core bandwidths. The mappings for TVOPD, TMPEG4 and NCS applications are depicted in Figure 5.2, Figure 5.3 and Figure 5.4, respectively. The core descriptions for each number identified in these mappings are described in the Table 5.1, Table 5.2 and Table 5.3 for TVOPD, TMPEG4 and NCS benchmarks, respectively.

Figure 5.2 – Mapping for HiCIT topology for TVOPD benchmark.



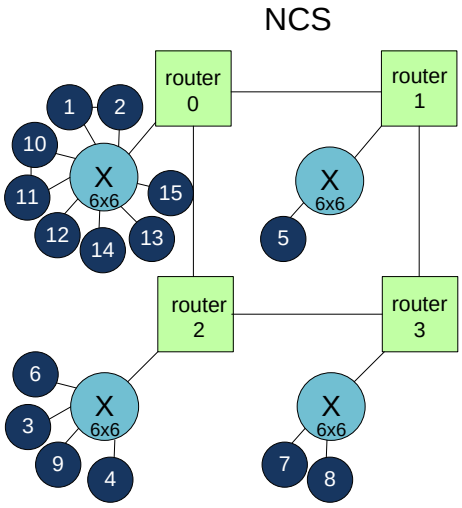
Source: elaborated by the author.

Figure 5.3 - Mapping for HiCIT topology for TMPEG4 benchmark.



Source: elaborated by the author.

Figure 5.4 – Mapping for HiCIT topology for NCS benchmark.



Source: elaborated by the author.

Table 5.1 - TVOPD benchmark core descriptions.

ID	core	ID	Core	ID	core
1	VLD0	14	VLD1	27	VLD2
2	RLD0	15	RLD1	28	RLD2
3	IQUAN0	16	IQUAN1	29	IQUAN2
4	IDCT0	17	IDCT1	30	IDCT2
5	ISCAN0	18	ISCAN1	31	ISCAN2
6	ACDC0	19	ACDC1	32	ACDC2
7	SMEM0	20	SMEM1	33	SMEM2
8	SAMP0	21	SAMP1	34	SAMP2
9	ARM0	22	ARM1	35	ARM2
10	VOPM0	23	MEM_OUT	36	VOPM2
11	PAD0	24	VOPM1	37	PAD2
12	VOPR0	25	PAD1	38	VOPR2
13	MEM_IN	26	VOPR1		

Table 5.2 - TMPEG4 benchmark core descriptions.

ID	Core	ID	Core	ID	core
0	SRAM1	12	VU1	24	AU2
1	VU0	13	AU1	25	MCPU2
2	AU0	14	MCPU1	26	RAST2
3	MCPU0	15	RAST1	27	SDRAM2
4	RAST0	16	SDRAM1	28	SRAM22
5	SDRAM0	17	SRAM21	29	ADSP2
6	SRAM20	18	IDCT1	30	UPSAMP2
7	ADSP0	19	ADSP1	31	BAB2
8	UPSAMP0	20	UPSAMP1	32	RISC2
9	BAB0	21	BAB1	33	IDCT2
10	RISC0	22	RISC1		
11	IDCT0	23	VU2		

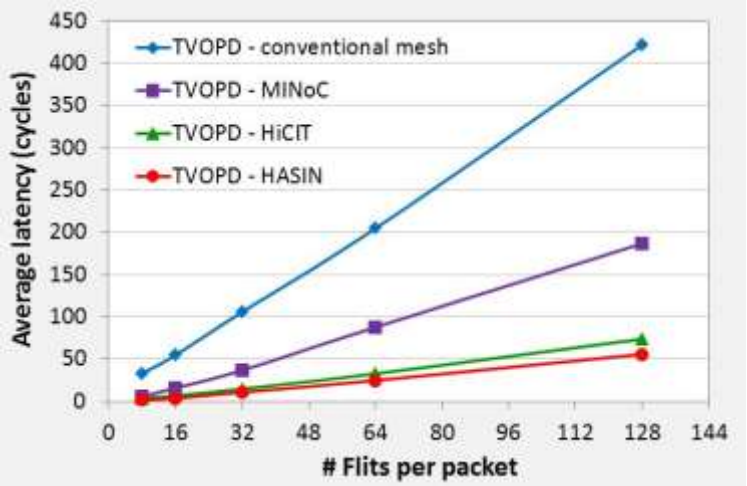
Table 5.3 - NCS benchmark core descriptions.

ID	core	ID	Core	ID	Core
1	SDRAM	6	UART	11	RAM3
	ARM	7	SW	12	EXT_IF
3	TIMER	8	RAM2	13	USB
4	ITC	9	RTC	14	DMA
5	ROM	10	ASIC	15	RAM1

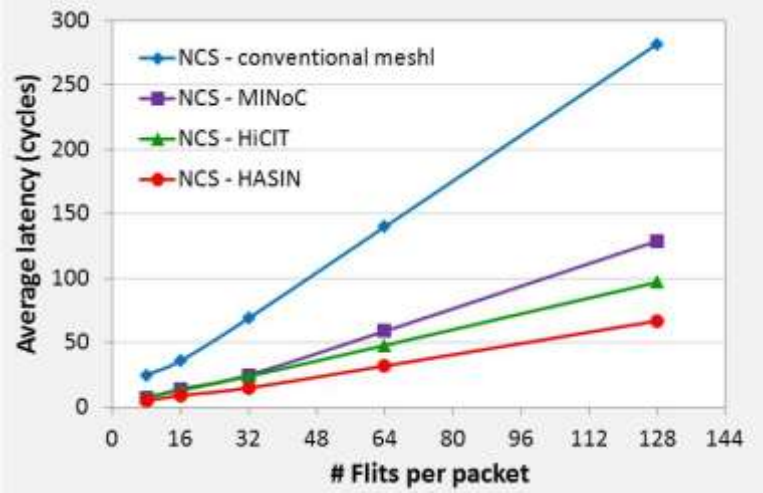
Initially, average latency results for the original benchmarks were analyzed. These results were obtained for the new mappings presented in Figure 5.2, Figure 5.3 and Figure 5.4 for HiCIT and HASIN topologies. The mesh mappings were obtained from ASHiNoC tool, as presented in chapter 3. Figure 5.5 presents a comparison between the network topologies. In this case, latency results for HiCIT, MINoC, HASIN and mesh NoCs were analyzed. According to these results, different analyses can be done. A reduction in the average latency was observed for all benchmarks when the two strategies presented in this work are employed (HiCIT + MINoC).

Figure 5.5 - Average latency results for HASIN in comparison with other strategies for (a) TVOPD, (b) NCS and (c) TMPEG4 benchmarks.

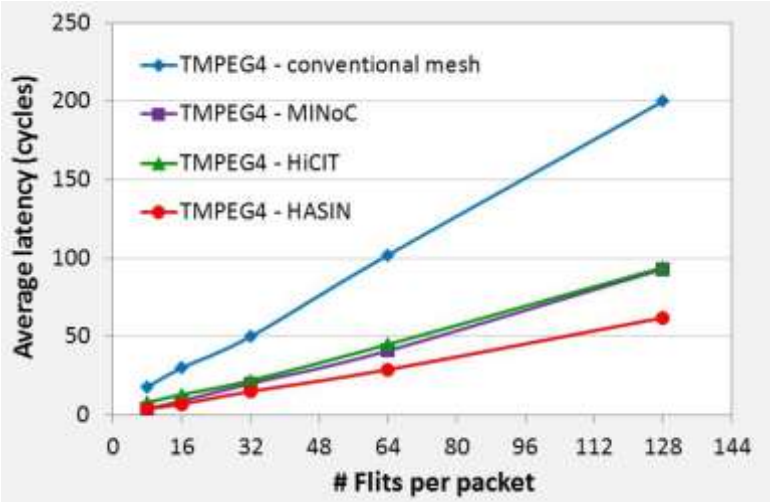
(a)



(b)



(c)



Source: elaborated by the author.

For the TVOPD benchmark, it can be observed that the average latency reduction was smaller than the other benchmarks. This is due to the behavior of this application. As presented in chapter 3, the TVOPD communication core graph presents a well behaved traffic, since there are few cores that receive messages of more than one. The most of the communication is given from a single core to another one, like streaming messages. Because of this, practically the mesh level does not present contention, and then, the adaptive strategy at this level does not obtain a considerable reduction in the latency. However, this situation is different for the NCS and TMPEG4 benchmarks, where the reduction in the average latency is expressive. The reductions in the average latency for the HASIN in relation to HiCIT range from 34% (with 128 flits per packet) to 46% (with 8 flits per packet) for TMPEG4 and, from 30% (with 128 flits per packet) to 37% (with 8 flits per packet) for NCS. If this comparison is made with a conventional mesh topology, the reduction is around 70% for the TMPEG4 and 76% for the NCS benchmark.

In the next subsections the importance of HASIN strategy whenever the communication rates of these applications suffer modifications will be verified. In the next experiments, in order to evidence the advantages in to integrate the hierarchy with adaptability, the results present the comparison with different strategies considering changes in the original application, as follow:

- When the application has a double bandwidth;
- When the application has a random mapping;
- When the application runs in a topology defined for other application.

5.1.2 HASIN Performance Results when the application has a double bandwidth

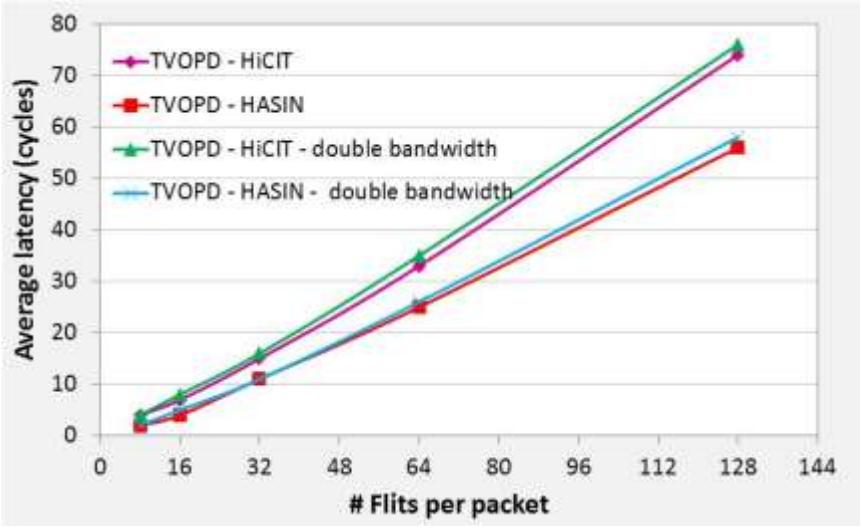
Another experiment was investigated in order to analyze the robustness of HASIN strategy. Although it was proved in the previous results the reduction in the latency for different applications using a hierarchical and adaptive NoC, the more important is to verify if this solution responses well when the bandwidth of some cores is increased or changed. In this case, to observe if the adaptability really mitigates the latency, all inter-cluster communications had their bandwidths doubled. The results for these traffic conditions are presented in Figure 5.6. In these results, it was compared the impact in latency when the HASIN is considered. If the percentage of increment in the latency is smaller for the HASIN

architecture than the hierarchical topology without adaptability, this proves that the purpose of the adaptive mechanism with a hierarchical topology was achieved.

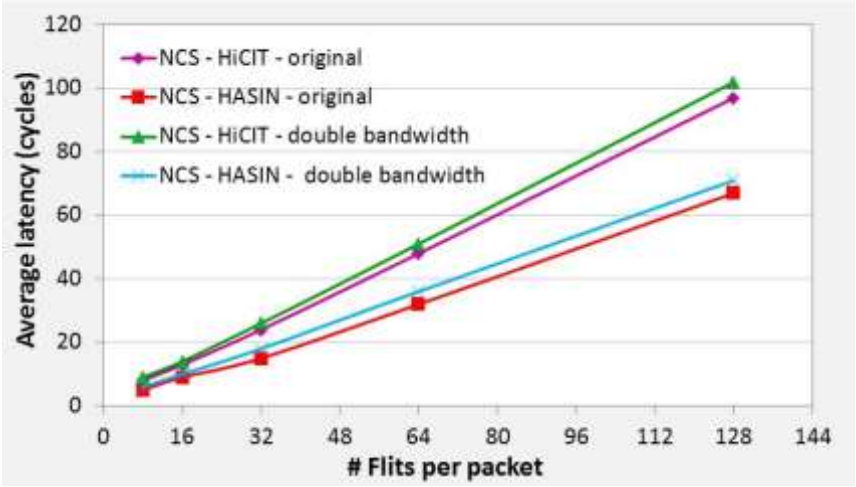
For the TVOPD and NCS, as the mapping for these applications greatly reduced the inter-cluster communications, the absorption of the latency impact was minimal for these cases. However, for the TMPEG4, as this benchmark presents more inter-cluster communications than the other benchmarks, the increment in the latency was minimized. These behaviors can be seen in the Figure 5.6 (c), where the difference in latency for the original communication rates compared to the doubled bandwidths is smaller in the HASIN proposal. In another experiment, besides to increase the bandwidths of the existing communications, more traffic was added in the global level, i.e., cores without communication started now to communicate. However, as the tool evaluates the average latency, and this is calculated from the total latency divided by the number of communications, increasing the traffic does not affect significantly the average latency.

Figure 5.6 - Average latency results for HASIN, HiCIT and MINoC when the bandwidths are changed for (a) TVOPD, (b) NCS and (c) TMPEG4 benchmarks.

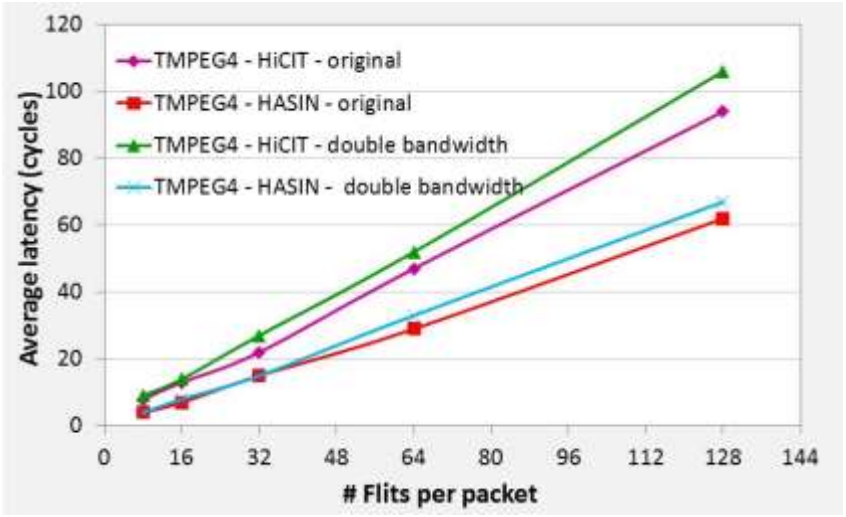
(a)



(b)



(c)

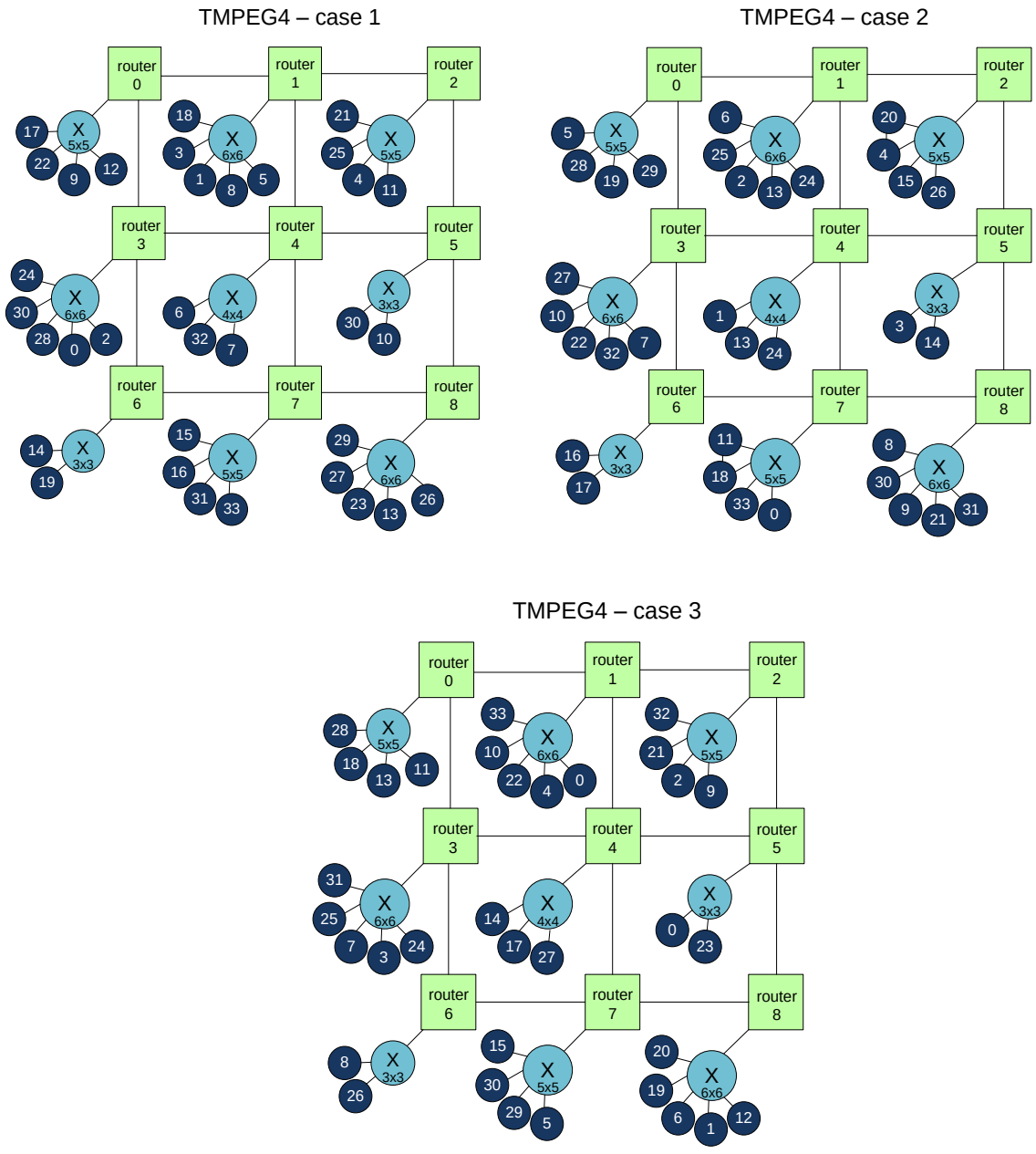


Source: elaborated by the author.

5.1.3 HASIN Performance Results when the application has a random mapping

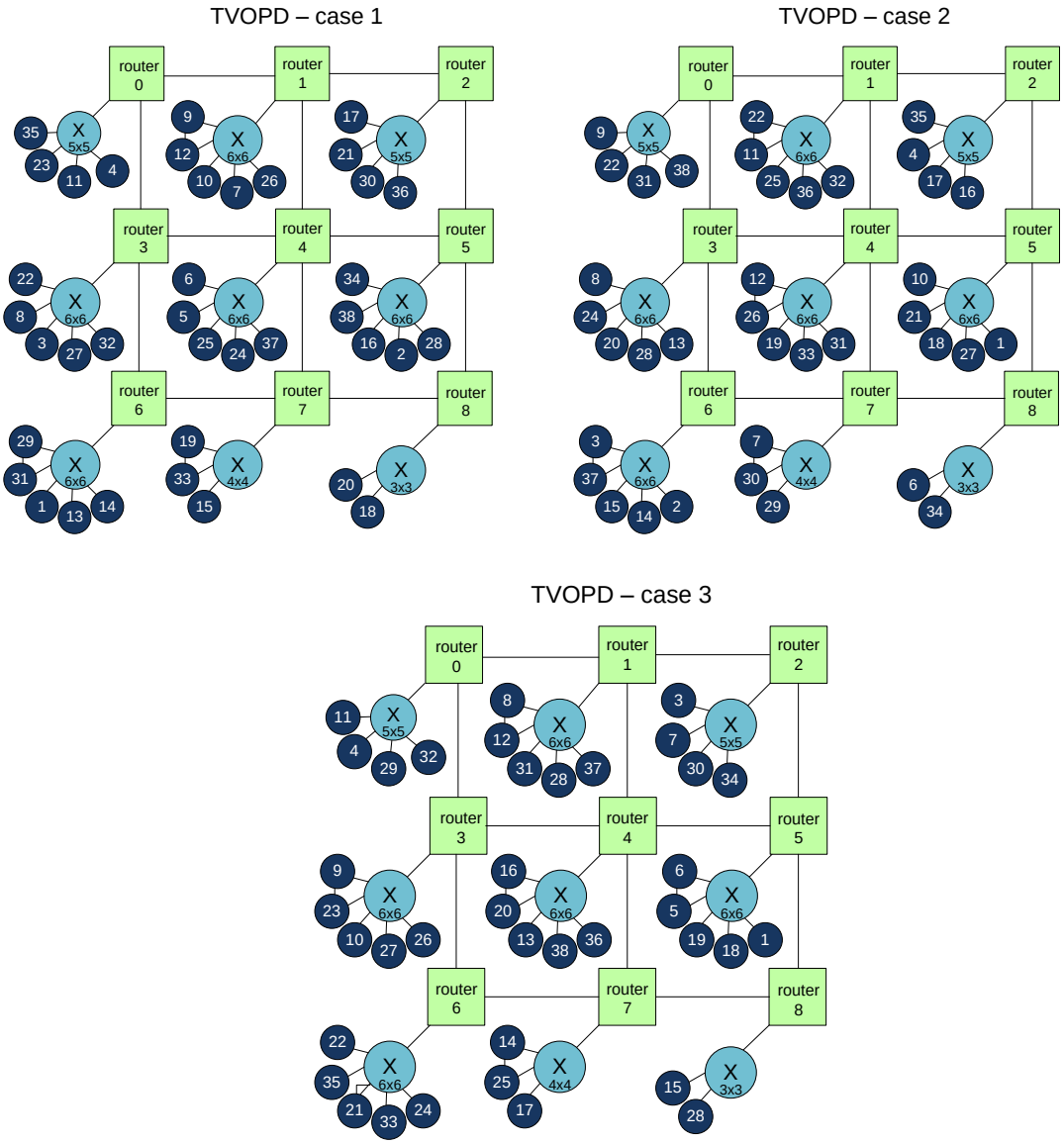
In these experiments, the impact in the results was verified when inappropriate HiCIT mappings are defined. For these analyses, 3 random mappings were verified for each benchmark. Random mappings for TMPEG4, TVOPD and NCS are presented in Figure 5.7, Figure 5.8 and Figure 5.9, respectively. The core descriptions according to the numbers indicated in the mappings were presented in Table 5.1, Table 5.2 and Table 5.3.

Figure 5.7 - Three random mappings for HiCIT topology for TMPEG4 benchmark.



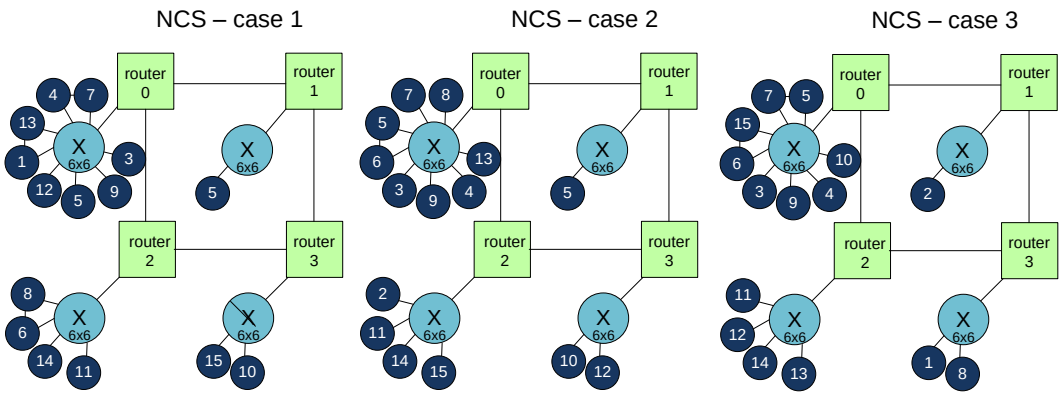
Source: elaborated by the author.

Figure 5.8 – Three random mappings for HiCIT topology for TVOPD benchmark.



Source: elaborated by the author.

Figure 5.9 - Three random mappings for HiCIT topology for NCS benchmark.

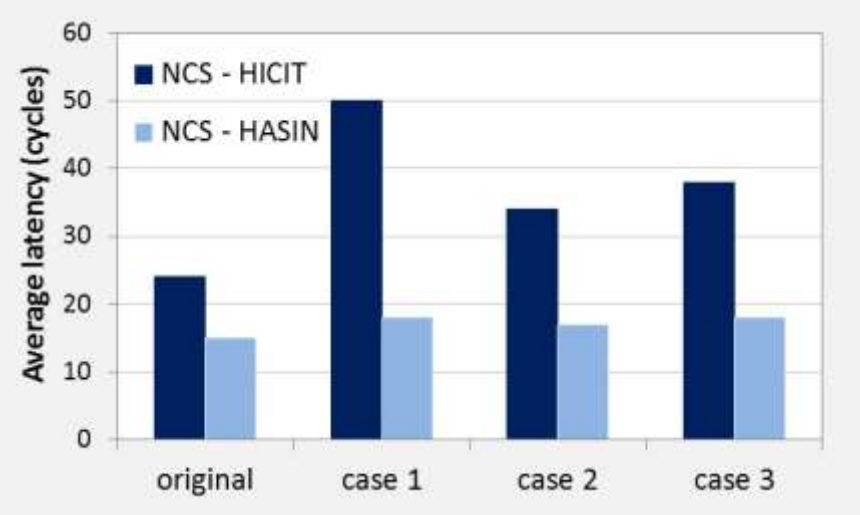


Source: elaborated by the author.

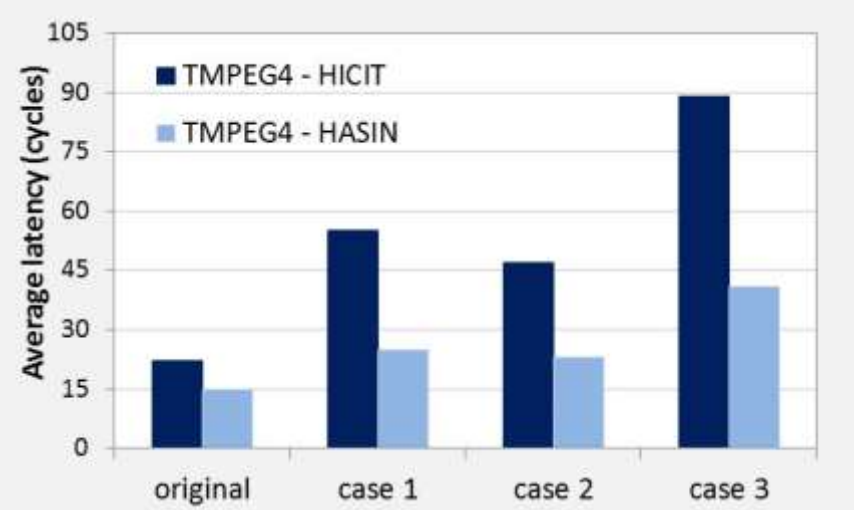
From these random mappings, it can be verified if the performance is impaired for HiCIT topology when the traffic changes completely. Besides of this, it can be seen if HASIN is able to minimize the impact in latency caused by the use of a hierarchical topology. Figure 5.10 presents the average latency for the random mappings. Due to the large amount of data, to better understand the results, only average latency for packets with 32 flits are illustrated. For the other packet sizes, the comparison between HiCIT and HASIN maintains a similar relation. For all reported cases, only the case 3 of TMPEG4 has presented an average latency worse than mesh NoC. This happens because all communications of this case occur in the inter-layer. However, this is an extreme case and very unlikely to occur from an initial optimal mapping. For the other cases, it is observed that some mappings present a large increment in the latency for HiCIT topology. However, the average latency for HASIN solution is mitigated when compared to the HiCIT. For the NCS application, the average latency for all cases is retained almost constant when compared to the original HASIN results. For the TVOPD and TMPEG4 benchmarks, the large reduction in the average latency using HASIN compared to HiCIT topology is viewable. For TVOPD, the average latency for the inappropriate mappings is increased in the HiCIT topology but it can be maintained smaller than 50% when using HASIN architecture.

Figure 5.10 - Results for HiCIT and HASIN topologies for random mappings for (a) NCS, (b) TMPEG4 and (c) TVOPD benchmarks.

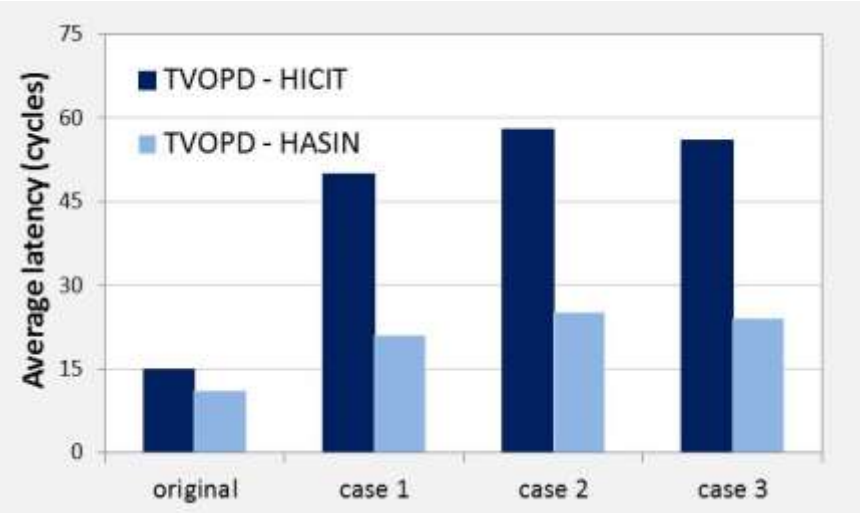
(a)



(b)



(c)



Source: elaborated by the author.

5.1.4 HASIN Performance Results when the application runs in a topology defined for other application

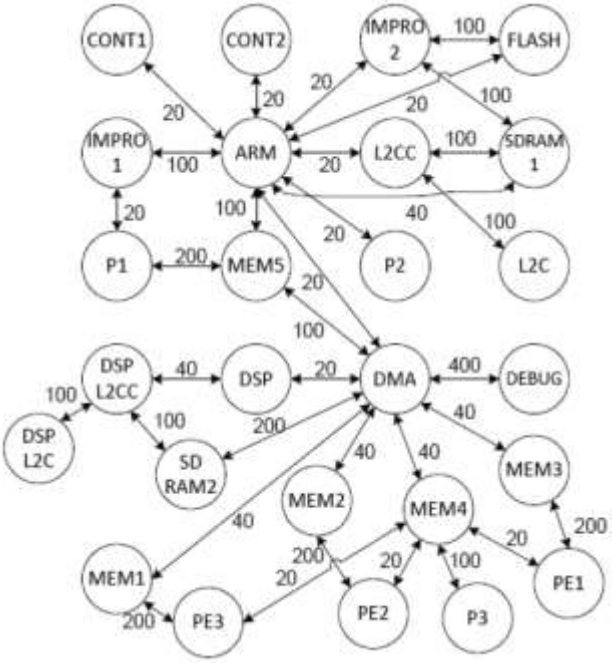
The last analysis related to the changes in the original traffic pattern was to adapt the mapping defined for an original application in another one. In this case, three situations were also considered. Other applications were verified in order to diversify the traffic conditions. These three cases are:

- Mapping for D26 benchmark in the TMPEG4 topology
- Mapping for NCS benchmark in the TMPEG4 topology
- Mapping for AV benchmark in the D26 topology

For this analysis, if one considers the cores as memories and processors, the new application needs to map the memories correctly. The core graphs for D26 and AV are presented in Figure 5.11 and Figure 5.12, respectively. The edges in the core graphs are represented in MB/s. D26 is a realistic multimedia and wireless communication. So, this benchmark contains 26 cores with irregular size and communication rates. The system includes ARM, DSP cores, multiple memory banks, DMA engine and several peripheral devices.

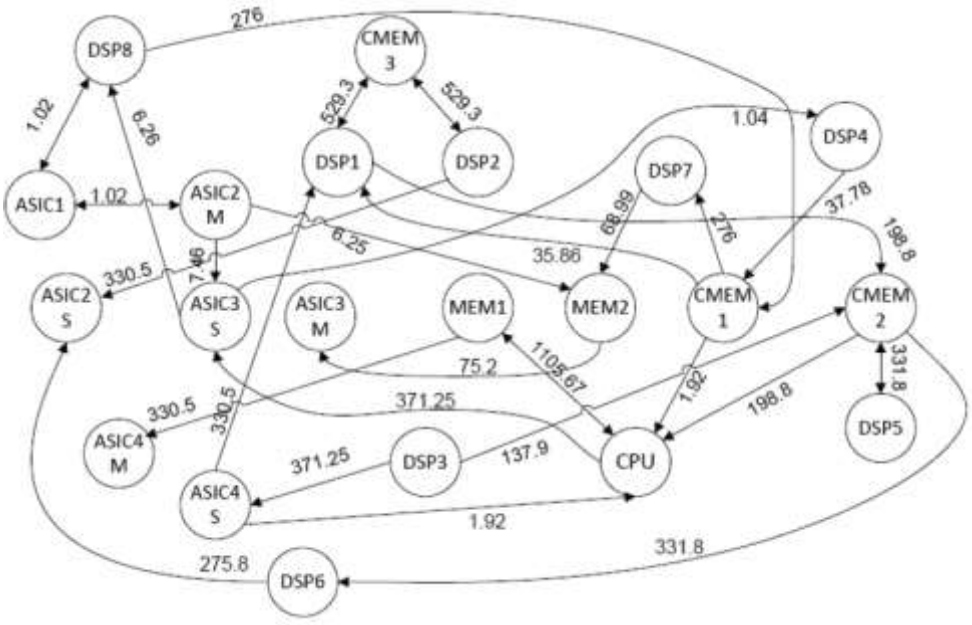
Audio/Video benchmark contains 21 cores and is used as a compression standard for digital audio-video (TINO, 2010). The original HiCIT mapping for D26 and AV benchmarks were defined according to (TINO, 2010). However, an interesting possibility was observed in these experiments: the possibility to maintain the cores originally mapped in a same cluster, forming also a cluster in another mapping. Therefore, this possibility was investigated and only when the cluster had a different number of cores or different memory allocation, the cores needed to be allocated in a different cluster.

Figure 5.11 - Core graph of D26 benchmark.



Source: elaborated by the author.

Figure 5.12 – Core graph of AV benchmark.

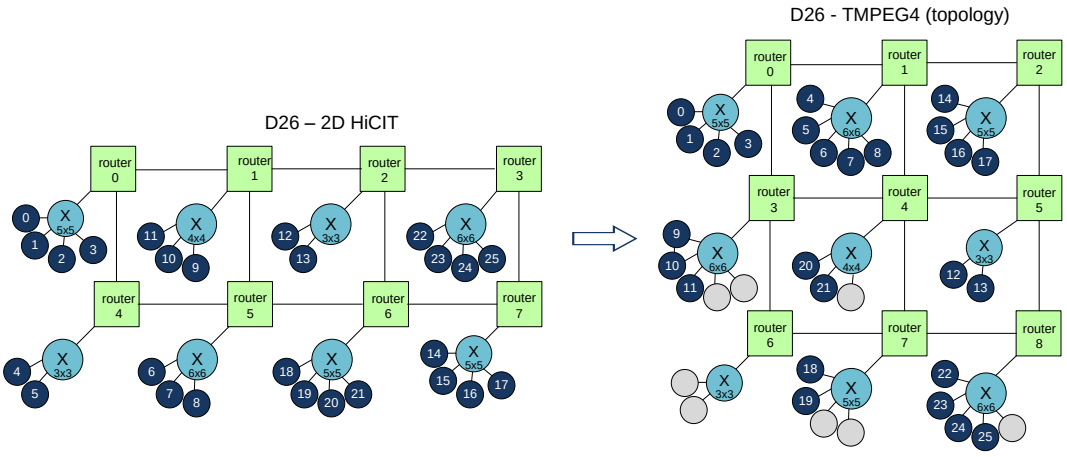


Source: elaborated by the author.

Figures 5.13, 5.14 and 5.15 present the original mappings and the adapted mappings for D26, AV and NCS benchmarks. The cores without identification in the clusters are not used in the new mapped applications. The corresponding core description for AV and D26 are presented in Table 5.3 and 5.4, respectively.

Figure 5.13 – (a) original mapping and (b) the mapping adapted for D26 benchmark from TMPEG4 topology.

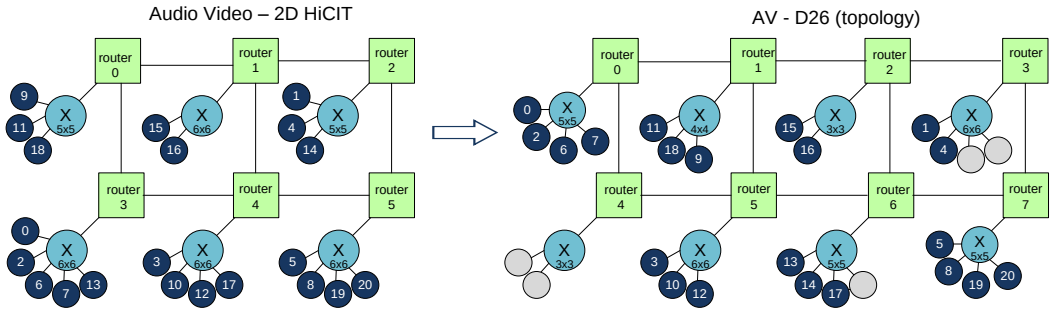
(a) (b)



Source: elaborated by the author.

Figure 5.14 - (a) original mapping and (b) the mapping adapted for AV benchmark from the D26 topology.

(a) (b)

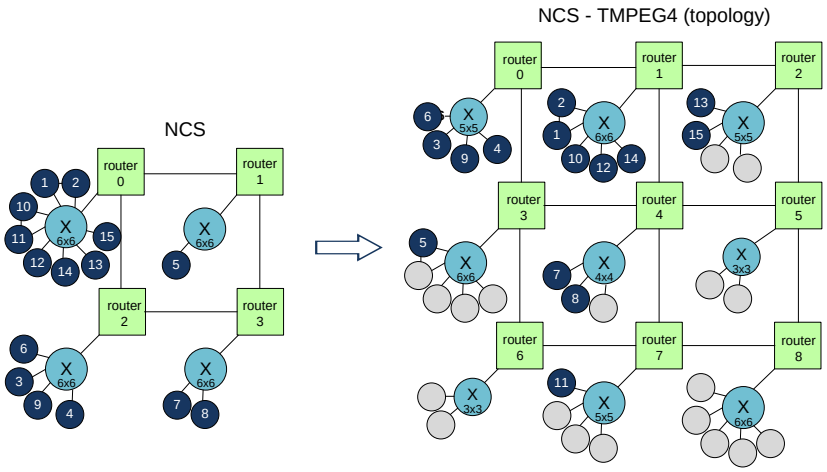


(a) (b)

Source: elaborated by the author.

Figure 5.15 - (a) original mapping and (b) the mapping adapted for NCS benchmark from the TMPEG4 topology.

(a) (b)



Source: elaborated by the author.

Table 5.4 - AV benchmark core descriptions.

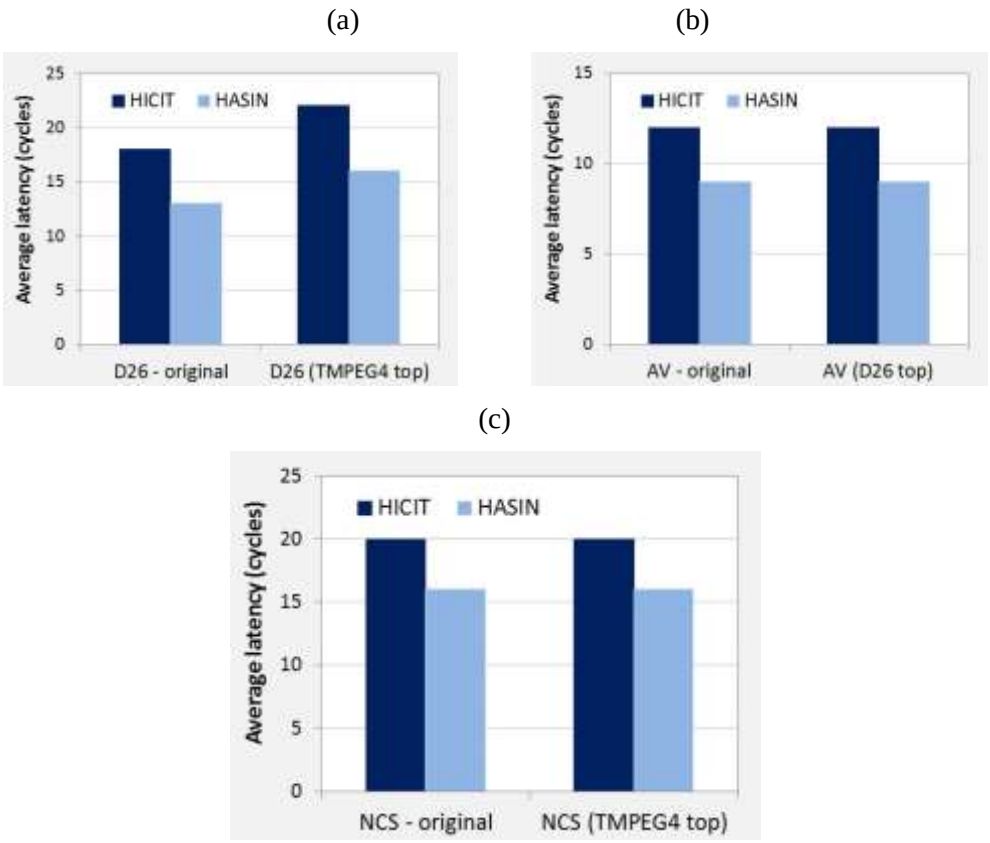
ID	Core	ID	core	ID	Core
0	DSP8	7	DSP4	14	CMEM2
1	CMEM3	8	ASIC2S	15	ASIC4M
2	ASIC1	9	ASIC3S	16	ASIC4S
3	ASIC2M	10	ASIC3M	17	DSP3
4	DSP1	11	MEM1	18	CPU
5	DSP2	12	MEM2	19	DSP5
6	DSP7	13	CMEM1	20	DSP6

Table 5.5 - D26 benchmark core descriptions.

ID	Core	ID	core	ID	Core
0	CONT2	9	L2C	18	MEM2
1	CONT1	10	SDRAM1	19	MEM3
2	IMPRO2	11	L2CC	20	PE3
3	FLASH	12	DAM1	21	MEM1
4	ARM	13	DEBUG	22	DSP
5	P2	14	MEM4	23	SDRAM2
6	IMPRO1	15	PE1	24	DSPL2C
7	P1	16	P3	25	DSPL2CC
8	MEM5	17	PE2		

The average latency results of these experiments are illustrated in Figure 5.16. As can be observed in the mappings, many cores were allocated exactly as the original mapping. Because of this, the average latency for HiCIT and HASIN were the same obtained for the original AV and NCS mappings. For the D26 benchmark, an increment in the latency was observed, but again HASIN is able to compensate the performance loss.

Figure 5.16 - Benchmarks mapped for topologies designed for other applications: (a) D26, (b) AV and (c) NCS.



Source: elaborated by the author.

5.1.5 HASIN Synthesis Results

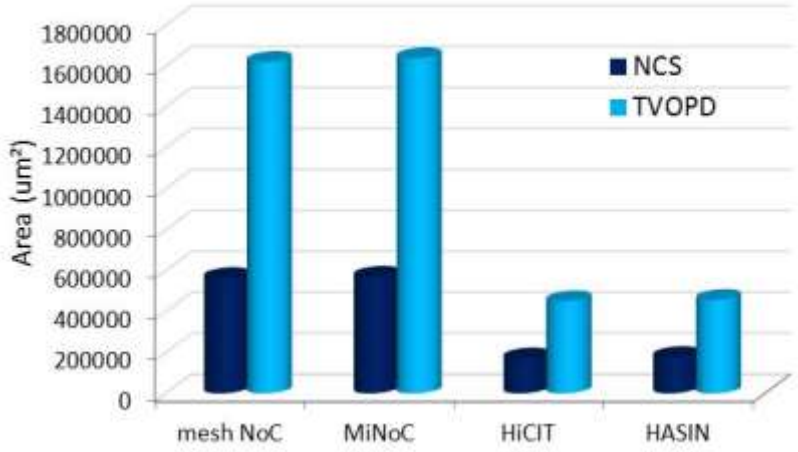
Synthesis results were obtained for the same configuration presented in section 4.4.2. The operating frequency considered in the synthesis was equal to 1GHz. Power consumption and area results are presented in Figure 5.17.

According to these synthesis results it is possible to observe that the integration of the hierarchical topology with adaptive strategy allows a large reduction in the power dissipation and area when compared to a conventional mesh NoC. The power and area reductions are still larger when compared to the MINoC proposal. This occurs because to provide the adaptability, MINoC requires the addition of more hardware than the conventional one. The reductions in area and power consumption for HASIN strategy compared to a conventional mesh NoC and MINoC are presented in the Table 5.6. One can conclude that the main gains of the HASIN proposal is to use the adaptability in a very low cost architecture like the interconnection using clusters composed of small crossbars combined with routers in a

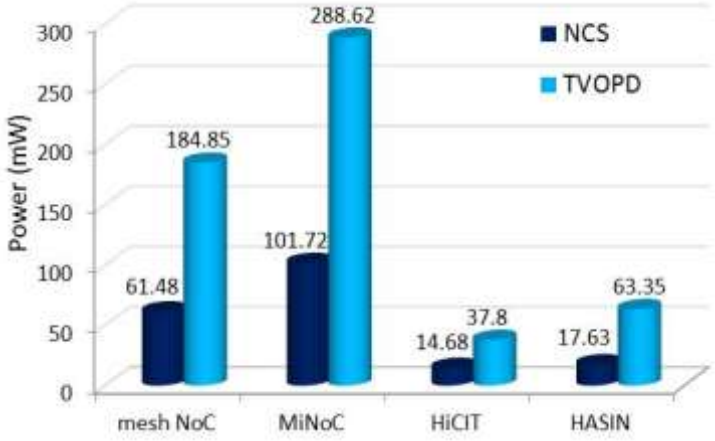
hierarchical manner. The reason related to the increment in the power consumption in the HASIN architecture compared to HiCIT is due to the MINoC architecture that presents a relevant increment in the combination logic, as already commented in subsection 4.4.

Figure 5.17 - (a) Power Consumption and (b) area results for different interconnection strategies for the NCS and TVOPD benchmarks.

(a)



(b)



(c)

Source: elaborated by the author.

Table 5.6 - Power Consumption and Area reductions obtained for the NCS and TVOPD applications with HASIN strategy.

		NCS	**TVOPD**
Power reduction (%)	HASIN x mesh NoC	71.32%	65.72%
	HASIN x MINoC	82.66%	78.05%
Area reduction (%)	HASIN x mesh NoC	68.46%	71.81%
	HASIN x MINoC	68.81%	72.12%

Although the power overhead is large in percentage when compared to HiCIT, it is important to observe that this comparison is related to a very simple NoC, that neither presents virtual channels nor tables for routing. Besides, if it is compared to other solutions with similar goals, such as (JERGER, 2008) and (MODARESSI, 2010), that present VCs and separated setup network, HASIN and MINOC would present a very small power consumption.

The increase in the power consumption when using HASIN instead of HiCIT proposal may not be desired or the application deadline is guaranteed operating in a lower frequency. In this case, one possibility is to use HASIN architecture operating in a lower frequency which allows reducing the power consumption. In this manner, latency parity was verified for HiCIT and HASIN considering the same benchmarks. HASIN at 500MHZ executing TVOPD and TMPEG4 have presented the same HiCIT latency operating at 1GHz. For the NCS benchmark, the latency parity is reached when HASIN is operating at 350MHz. Table 5.7 presents the power consumption when the architectures operate with these frequency values. As can be verified from Table 5.7, a considerable power reduction can be obtained with the latency parity for NCS and TMPEG4. However, as the TVOPD in the HASIN topology consumes even further power consumption than HiCIT, reducing the operating frequency was not be able to reduce the power consumption in this case, but the HASIN solution will still present the adaptability advantage.

Table 5.7 - Latency parity for HiCIT operating at 1GHz and HASIN operating at 500MHz.

		Power Consumption (mW)		
	Benchmarks	TVOPD	TMPEG4	NCS
HiCIT	1GHz	37.80	33.39	14.68
HASIN	500MHz	46.45	27.01	-
	350MHz	-	-	9.38
Power reduction (%)		-22.9%	19.1%	36.1%

5.2 Considerations

In this chapter was presented an efficient solution that integrates adaptability with hierarchy. The results evidenced that this solution absorbs the increment in power consumption occasioned by the adaptability strategy. Besides, the HASIN architecture obtains a large improvement in performance. The advantages of this strategy were possible due to the hierarchy that allows exploring the communication locality combined with a dynamic

adaptability strategy. As one can observe, only the use of the adaptive switching strategy is not enough to reach good results in both power and performance. It is clear the use of different techniques need to be considered to increase the performance and reduce the costs.

6 THREE DIMENSIONAL NETWORKS-ON-CHIP

As the complexity of the integrated circuits has grown exponentially over the last years, and despite previous proposals presenting a large improvement, an ideal interconnection solution for large system in a single layer remains a challenge (SHEIBANYRAD, 2010), (KUMAR, 2012). Different 2D NoC proposals have been analyzed for interconnecting these systems, such as adaptability in NoCs, new topologies, virtual channels and others. However, the use of a conventional 2D NoC is complicated by the increase in the number of PEs on a chip due to the latency of transmitting the messages (many hops to reach the destination) and the difficulty to implement in the same layer, elements designed for distinct technologies (PASRICHA, 2009), (WEERASEKERA, 2009). From this emerged the 3D integration, where multiple silicon layers are stacked vertically, allowing wire length reduction. Thereby, the long horizontal wires in a 2D design are replaced by shorter vertical links, which favor faster on-chip communication (DAVIS, 2009), (CHAO, 2010), (KUMAR, 2012). In this way, three-dimensional integration is perceived as a solution to scale the performance of electronic devices beyond Moore's law. 3D integration by stacking dies on top of each other and interconnecting them with Through Silicon Vias (TSVs), achieves a drastic reduction in wire length with clear benefits in performance (DAVIS, 2009), (CHAO, 2010), (SEICULESCU, 2009).

3D integration on the other hand, achieves device density multiplication by stacking IC layers without aggressive scaling. Therefore, this solution may be a viable and immediate remedy as conventional scaling becomes less cost effective (TAN, 2011). Besides, 3D designs allow the use of mixed-technologies stacked in different layers, e.g. digital, analog, memory, RF, Micro Electro-Mechanical Systems (MEMS), thereby offering a great opportunity to diversify the functionality of electronic devices (DE PAULO, 2010). Other advantages gained from the 3D integration are the noise isolations among analog/ RF and digital circuits. However, 3D Integrated Circuits (3DIC) technologies also bring about new challenges. Stacking layers, interconnected by a full-interconnected 3D-NoC considerably increases area and power due to the TSV interconnections, as will be detailed in the next section.

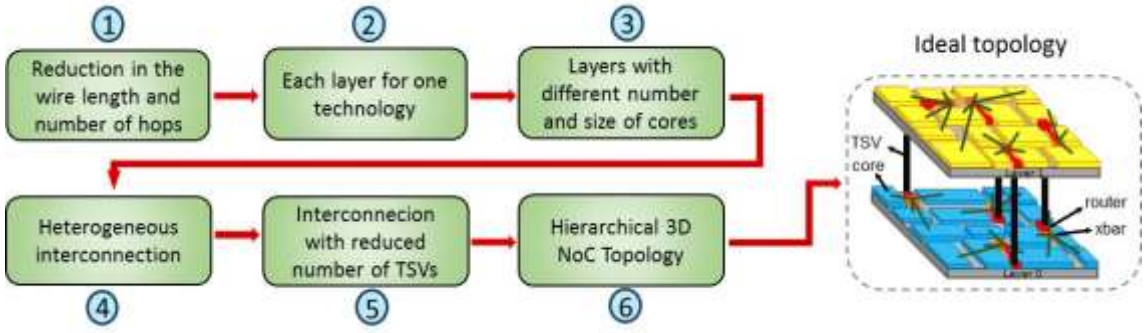
Over the past years, NoCs were studied as an appropriate solution for complex hardware systems due to scalability, parallelism and quality-of-service (DALLY, 2001), (BENINI, 2002). However, 3D integration presents some specific features and peculiarities that require attention. NoCs for 3D designs need to have extra channels to be interconnected to the TSVs and therefore some analyses need to be considered in order to achieve the aforementioned advantages obtained from these designs. Depending on TSV fabrication technology, the project of the vertical pillars can present some constrains, as their pad requires large area and power (VELENIS, 2009) (XU, 2011).

In this chapter the techniques previously applied for 2D NoCs in the context of 3D integration are analyzed. The same strategy based on crossbar-switch clusters is considered. The eventual communications between clusters is made on a higher hierarchical level by a mesh packet-switching NoC. The difference is now the router which is designed for 3D interconnection. Although it seems that the previous solution was directly applied in the context of 3D NoCs, in fact several studies had been considered. From these analyses, one concludes that the proposed hierarchical topology adapts well for the concept of 3D interconnection. In the next subsection, the 3D interconnection scenario will be contextualized, its challenges and the topology that favor the gains of multi-layer designs.

6.1 3D NoC paradigms

In this section, many factors that impact directly in the benefits of 3D designs will be discussed. Many reasons have encouraged the development of multi-layer chips; however, it has been proven that there are specific scenarios where this architecture is more suitable. At least six important points about 3DIC need to be observed and these are illustrated in Figure 6.1.

Figure 6.1 - Six points which led the ideal interconnection solution.



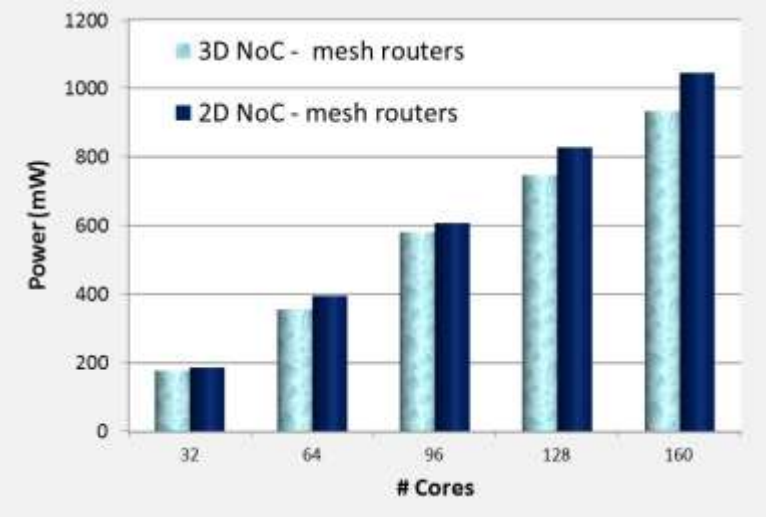
Source: elaborated by the author.

6.1.1 Reduction in the wire length and number of hops

The first advantage largely evidenced and reported in different works related to the three dimensional integration in comparison with a 2D design, is the reduction in the wire length (FEERO, 2009) (WEERASEKERA, 2009), (PASRICHA, 2009), (KUMAR, 2012), (RAHMANI, 2012). Two-dimensional architectures designed for large systems, composed of numerous cores, result in several bottlenecks due to the routing congestion (FEERO, 2009), (WEERASEKERA, 2009). Besides, an interconnection with one more dimension, it is possible to reduce the number of hops from source to destination (PAVLIDIS, 2011). On top of this, interconnection delay of these circuits has become an important challenge of system performance (PASRICHA, 2009) and one alternative which allows continued progress is the use of 3D integrated circuits. Wire power dissipation can be an important advantage in 3D NoCs. Figure 6.2 presents the results for meshes, increasing the topology size and the number of cores for 2D and 3D designs. For this experiment, cores with 1mm² as black box in the synthesis to obtain the correct wire power were considered. As can be observed, 3D NoCs reduces the power consumption when compared to 2D NoCs.

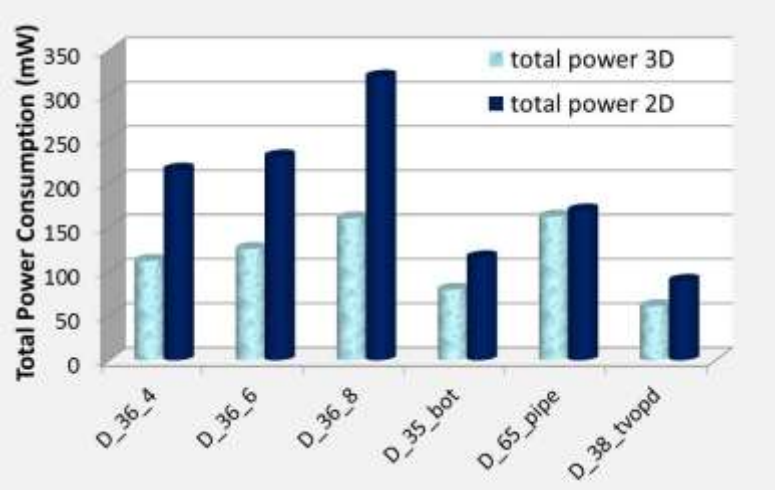
In (SEICULESCU, 2009) a comparison of different benchmarks considering 2D and 3D NoCs also presents saving power in interconnections for stacked circuits, as illustrated in figure 6.3. In the experiments of this paper, data width of the NoC links presents 32 bits, the NoC operating frequency is 400 MHz and the libraries are low power for 65nm. The results in Figure 6.3 point out larger gains in the power consumption than were found in the results of Figure 6.2. This difference occurs due to the difference in the topologies, since in (SEICULESCU, 2009) switches were used to interconnect many cores, and in tools and libraries used in each experiment. It is important to note that the power of the switches is increased in the 3D designs, but in some cases, this overlap is compensated by the reduction of power due to the shorter wires and removal of repeaters. Although the total power consumption of a 3-D system is lower than that of an equivalent 2-D circuit, 3D integrated circuits exhibit a profound increase in power density due to the reduction of chip size (PAVLIDIS, 2011). In this manner, another obstacle in the 3D integration is the increase of the chip temperature because of its poor thermal conductivity and heat dissipation (BANERJEE, 2001).

Figure 6.2 - Power consumption for 3D and 2D mesh NoCs for different number of cores



Source: elaborated by the author.

Figure 6.3 - Comparison between 2D and 3D NoCs for different benchmarks.



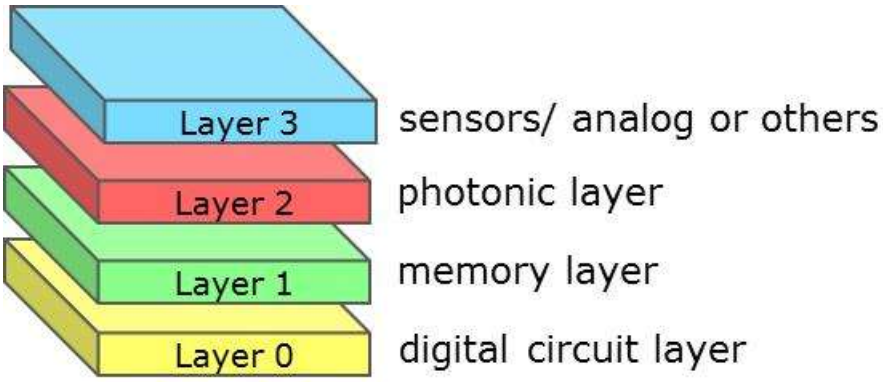
Source: SEICULESCU (2009).

6.1.2 Possibility to consider one layer for each technology

Another important issue is the possibility of integrating disparate technologies in the layers, as illustrated in Figure 6.4. In mixed-signal systems, the circuitry is prone to failure due to interferences. If the technologies are in different substrate then noise isolation between analog/RF and digital circuits is avoided (WEERASEKERA, 2009). However, there are several issues with regard to 3D circuits since the research in 3D integration is relatively new. One challenge is to understand how the technologies will be integrated and how intense will be the communication between the layers. These answers are very important to define the interconnection architecture. Apart from the technologies that distinguish it from digital, one

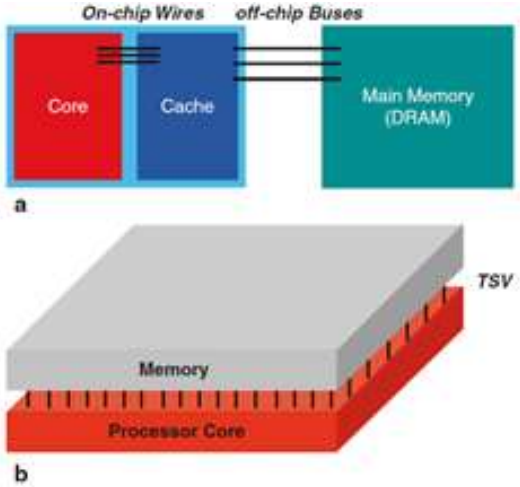
trend in multi-layer chip is to dispose the memory in a specific layer to communicate with the processing elements (TAN, 2009), (WELDEZION, 2009), (SEICULESCU, 2009). One clear possibility for memory hierarchy in 3D, is the integration according to Figure 6.5 since it allows increasing the memory bandwidth (TAN, 2009). The use of bus between the layers has been considered by some authors (LI, 2006), (SUN, 2009), (RAHMANI, 2011). This strategy can be a good solution for 3D memory hierarchy communications, but it can be a bottleneck if many cores require access to the same layer, since the buses do not allow concurrent communication, which in a high network load can critically increases contention (RAHMANI, 2012).

Figure 6.4 - Example of specific technologies for each layer of a 3D integration.



Source: elaborated by the author.

Figure 6.5 - (a) Memory hierarchy and (b) its disposition in a 3DIC. .



Source: TAN (2009).

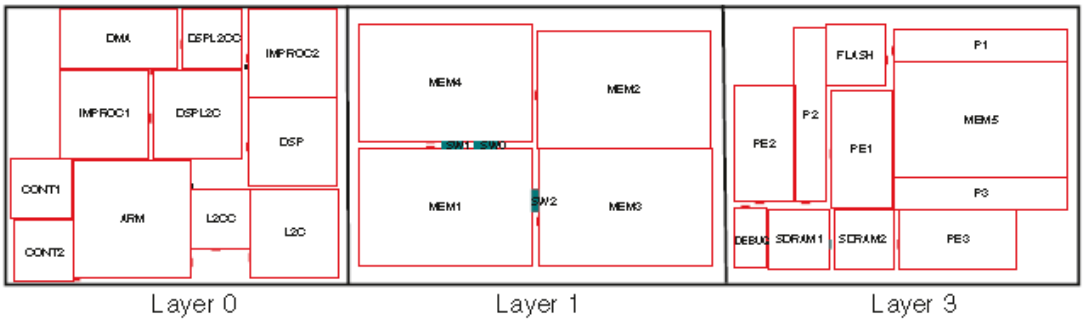
6.1.3 Layers with different number and size of cores

Assuming cores completely different in each layer, designed for a specific technology, it is impossible to consider a homogenous interconnection solution, as for example a fully

interconnected 3D mesh NoC for these systems. Moreover, even if the technology in the layers is the same, as the PEs operate different functions and have different architectures, the system still cannot use a homogeneous interconnection device since the number and area of the cores are different between the layers. Two examples of this reality are demonstrated in the floorplans for D26 (SEICULESCU, 2011) and TVOPD (MURALI, 2009) benchmarks in figure 6.6 and figure 6.7, respectively. In layer 1 of the floorplan for D26 benchmark for example, only 4 memories are integrated. In layers 0 and 3 a larger amount and smaller cores are mapped. This same feature is seen in the TVOPD benchmark, proving how inappropriate it would be to use a regular topology to connect the layers.

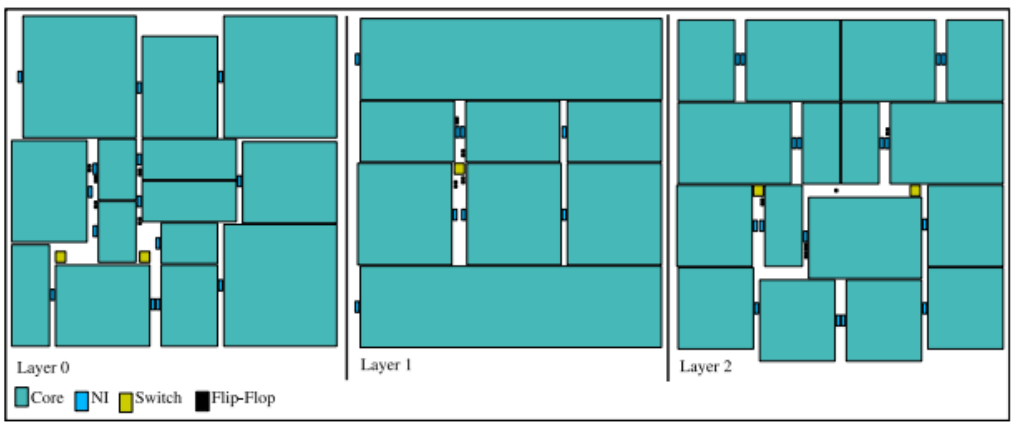
In both examples, Figure 6.6 and 6.7, switches were used to interconnect the cores. These switches are similar to routers; however they present channels with buffers and controls which substantially increase the area and power. In (SEICULESCU, 2011), for example, large switches were defined in the topology. Besides, in the floorplans of figure 6.6 and 6.7, dedicated TSVs were added to interconnect cores of distinct layers. Actually this is not the best strategy since TSVs are expensive and ideally topology where they can be shared by many cores is adopted.

Figure 6.6 - 3D floorplan for D26 benchmark.



Source: SEICULESCU (2011).

Figure 6.7 - 3D floorplan for TVOPD benchmark.



Source: MURALI (2009).

6.1.4 Heterogeneous interconnection

Continuing the discussion of the last issue, if a homogeneous 3D NoC topology is applied to interconnect heterogeneous cores, as for example, the cores of the D26 and TVOPD applications, the asymmetry of the layers will hamper the design of vertical links for all routers. Besides this, unnecessary extra resources would be added, for example routers without a PE attached. In this case, extra routers are used only to complete the topology corresponding to the other layers. Moreover, there is probably an increase in communication latency, since many hops may be required to reach the destination core. Each flit must undergo buffering and arbitration at every hop, adding to the overall delay (XIE, 2010). Thus, ensuring low power and high performance for heterogeneous systems: heterogeneous interconnections. As previously discussed, being heterogeneous systems 3D designs favor the integration of different technologies. And moreover, three-dimensional integration aids noise isolation (WEERASEKERA, 2009), (KUMAR, 2010).

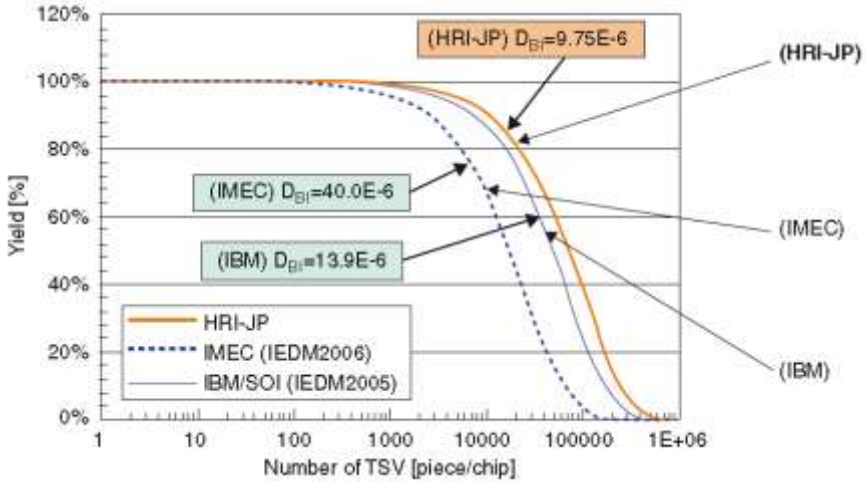
6.1.5 Interconnection with reduced number of TSVs

Stacking multiple dies vertically is not new. Other strategies have already been proposed in the past, such as System-in-Package (SiP) and Package-on-Package (PoP) (XIE, 2010). Nevertheless, recent studies point out the TSV as a viable solution to build vertical interconnections in 3D chips (XU, 2010), (XU, 2011). However, this additional infrastructure has to be considered along with the silicon area occupied by TSVs, which despite technological advances, are not likely to be below 1□m diameter (ITRS, 2011). Besides, the manufacturing process of the TSV is expensive and delicate (VELENIS, 2009). The extra fabrication step for TSVs and die-bonding are prone to defects, requiring redundant connections and built-in test structures (PASCA, 2012). In this way, depending on the technology, the yield of the chip die reduces with the number of TSVs due to the fact the TVS process incurs a larger probability of defects compared to the traditional 2D designs, such as misalignment, random defects, additional of manufacturing process (AKBARI, 2012), (SEICULESCU, 2011). TSVs also occupy significant silicon area; however, previous research on 3D placement and routing did not consider this fact (KIM, 2009). Many strategies are not realistic when considering the possibility of implementing thousands of TSVs (GOPLEN, 2007).

In a regular topology, it is clear that the maximum performance is achieved through a full interconnected network-on-chip, i.e., all routers presenting vertical up/down connections to the pillars without using serialization in these links. Nevertheless, it is hard to ensure this manufacturing, due to the costs and chip area. According to (XU, 2011), for a 65nm of CMOS technology, the costs with TSV processing are around of 46% to 65% of the wafer manufacturing costs. Pasricha considered some analyses about the TSV area in (PASRICHA, 2009). For example, in a chip with a hundred 64-bit vertical TSV links (expecting MPSoCs with hundreds of cores), TSVs with pad dimensions of 10umx10um and a pitch of 16um will occupy an area equivalent to the size of one processor.

Figure 6.8 presents how the yield for different process varies with the number of TSVs used across two layers (SEICULESCU, 2011). The graphs show a trend that upon reaching a certain threshold, the yield starts to decrease with increasing number of TSVs. This threshold is approximately 1000 TSVs. In this figure, TSV is considered as the set of vias defined for each vertical interconnection. Each TSV process presents a DBI (Direct Bond Interconnect) value. Vertical interconnection DBI is a composition of Cu/oxide hybrid bonding technology (ENQUIST, 2006). The surfaces of non-conductive oxide and conductive Cu are aligned and placed together resulting in a bond.

Figure 6.8 - Number of TSVs x Yield.



Source: SEICULESCU (2011).

Due to all these factors, the number of TSVs needs to be reduced to minimize the number of faults, as well as reducing the power and area results, but it is important that this does not compromise the system performance.

Some research presents solutions to avoid the relevant problems caused by designs with too many TSV pillars. One alternative is to serialize the TSV interconnections, as

proposed by (PASRICHA, 2009), (PASRICHA, 2012). However, with this strategy, the system will present a loss in performance. Another solution is to reduce the number of pillars and define an appropriate placement in order to ensure performance (XU, 2010), (XU, 2011). Some other works still consider the conventional mesh topology with a reduced number of vertical interconnections and propose an appropriate routing algorithm to define the path through vertical links (DUBOIS, 2013), (YIN, 2012). Even so, multiple cores will need to share the same vertical links, increasing the congestion of the network and adding communication bottlenecks. However, a tight TSV constraint would force fewer inter-layer links, thereby increasing congestion on such links and affecting performance (SEICULESCU, 2011).

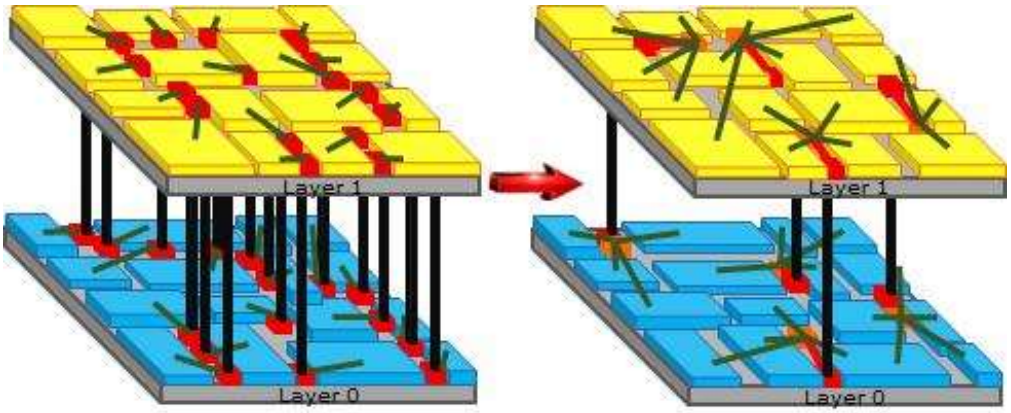
6.1.6 Hierarchical 3D NoC topology

The question related to the reduction in the number of TSVs is: how to ensure low latency combined with low power NoCs for 3DIC? The conclusion is that the ideal strategy is to explore the communication locality in the network topology, since these features have become even more important in 3D chips. If the communications are concentrated, it does not justify a fully vertically interconnected 3D NoC. Allied to this, the interconnection needs to meet also the last points, considering heterogeneous interconnection since the cores are heterogeneous in terms of size, function, bandwidth and communication, and reducing the number of TSVs in the vertical links. All these arguments point out strong reasons to consider a hierarchical 3D NoC topology. From these points, it is evident that the benefits of the HiCIT topology also apply to 3D designs. In 3D integration, a HiCIT router has 7 ports (two additional ports for up and down interconnections). With this solution it is possible to largely reduce the number of pillars compared to a conventional mesh topology, as can be observed in Figure 6.9.

With regard to 3D hierarchical NoC topologies, there are a few proposals in the literature. The majority of studies focus on 3D mesh NoCs, assuming 7x7 routers (DE PAULO, 2010), (EBRAHIMI, 2011) or topologies composed of a combination of routers with 6 ports and a bus for the vertical interconnections (DANESHTALAB, 2010), (RAHMANI, 2012), (LI, 2006), (SUN, 2009) but with their respective TSV pillars for each router and one router for each PE. Many 3D-NoC proposals do not consider all these relevant issues, which, in fact presents a realistic analysis regarding future requirements of MPSoCs. There is no hierarchical 3D NoC proposal that presents the same important features of our architecture. In

(LAFI, 2010), a hierarchical router composed of two routers was proposed, one 5x5 router (like a conventional 2D mesh router) and one 4x4 router (for vertical interconnection). However, this proposal does not tackle the problems related to the number of TSVs in the chip, neither presents power advantages, as our solution does. In (VISWANATHAN, 2012), two other hierarchical 3D NoCs were analyzed, a 3D star topology and a 3D recursive network topology. However, the authors do not mention the architecture of these proposals and besides this; a specific and more complex routing algorithm needs to be implemented for these topologies.

Figure 6.9 - A comparison related to the number of pillars required for a (a) conventional 3D mesh topology and (b) the proposed hierarchical 3D topology.



Source: elaborated by the author.

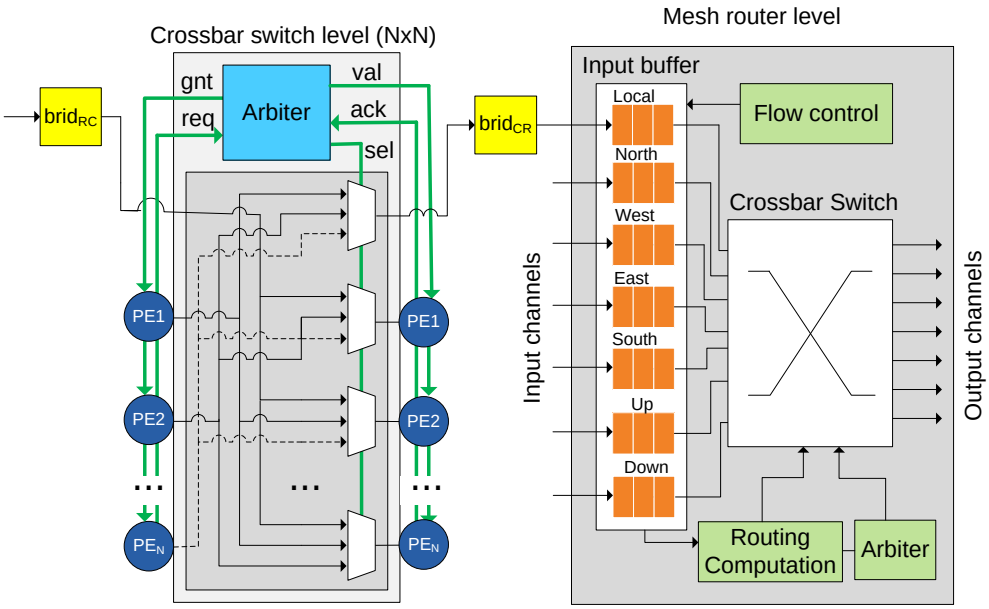
6.2 Proposed Hierarchical 3D NoC

6.2.1 Hierarchical Architecture

The proposed solution for 3D integration will from here on be referenced as 3D-HiCIT (Three Dimensional Hierarchical Crossbar-based Interconnection Topology) since the idea is the same as presented in the hierarchical topology for 2D circuits. However, in this case the router is designed with seven ports, six to connect to the neighboring routers (North, South, West, East, Up and Down) and one to connect to the local level, instead of attaching directly to a PE as occurs in a conventional mesh topology.

With 3D-HiCIT, the hierarchical interconnection proposal allows a substantial reduction in the number of 3D routers and consequently in the number of TSVs. The architecture for 3D-HiCIT, as presented similarly in chapter 4, is presented in Figure 6.10.

Figure 6.10 - 3D-HiCIT architecture.

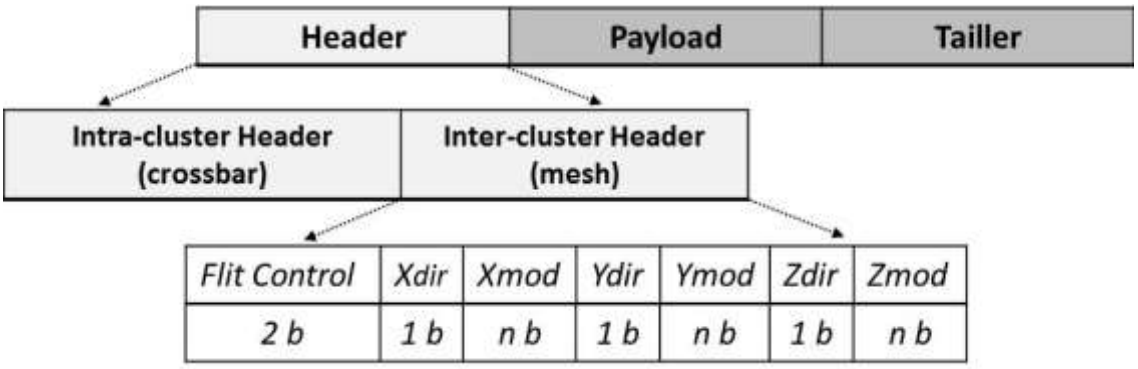


Source: elaborated by the author.

6.2.2 Packet Format

The packet format for 3D-HiCIT is presented in Figure 6.11. The routing algorithm for the top level can be any strategy already considered in the mesh topologies. In this work the XYZ was used. In this case, for each dimension 1 bit was used to define the direction and n bits to define the number of hops to the destination cluster.

Figure 6.11 - 3D-HiCIT architecture.

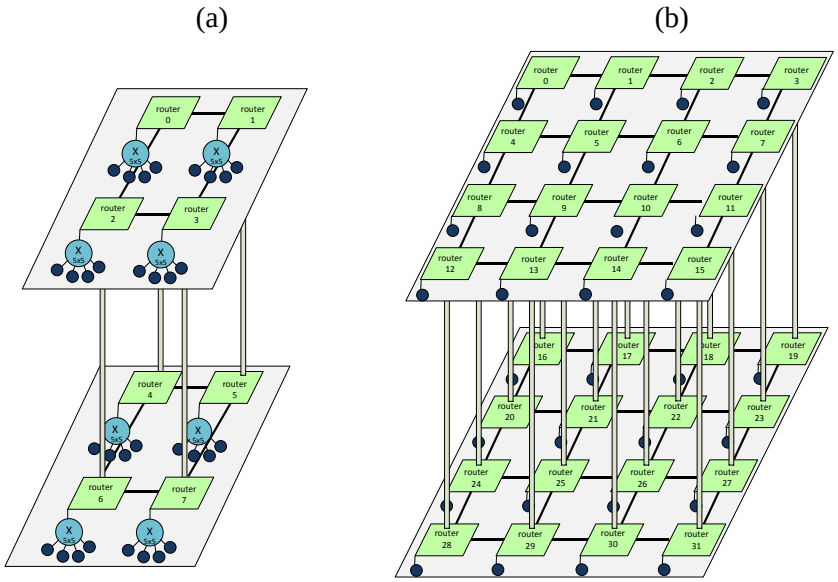


Source: elaborated by the author.

6.3 Experimental Results

Synthesis results were analyzed for the 3D-HiCIT topology and compared to a 3D mesh topology, as illustrated in Figure 6.12. In this figure only the routers are emphasized, so the cores did not represent a proportional size compared to the routers. The results were also compared to 2D-HiCIT and 2D mesh NoC.

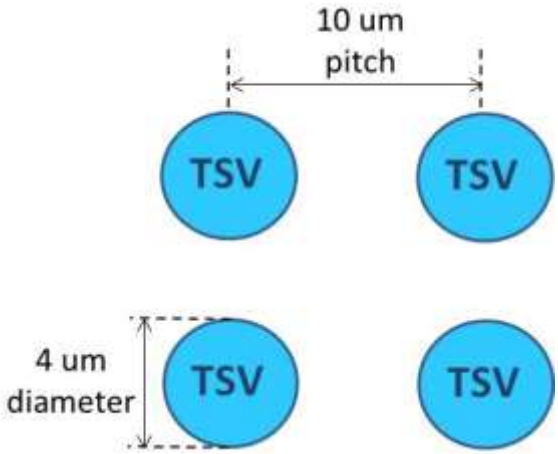
Figure 6.12 - Example of (a) 3D-HiCIT and (b) 3D-mesh topologies for 32 PEs.



Source: elaborated by the author.

For the synthesis results, the same tools previously adopted (Cadence RTL Compiler for the logical synthesis and Cadence First Encounter for the physical synthesis) were considered. As these tools are for 2D designs, the area and power results were estimated according to the flow charts of Figure 6.14 and Figure 6.15, respectively. In this case, for the area, it was need to obtain the total gate area of the design (with the Cadence RTL Compiler), since the routers present 7 ports. In order to obtain the wire area, synthesis for each layer was required, considering the First Encounter tool after the synthesis with RTL Compiler tool. The total wire area is the sum of wire area of each layer. Finally, the total area is the sum of total wire area plus gate area and plus TSV area. As can be verified, the total area is not the chip area but the area related to gates, wires and TSVs. The TSV area estimation considers 4um diameter and 10um pitch, according to illustrated in Figure 6.13. Pitch represents the distance between two adjacent vias. This TSV diameter and pitch are realistic values for 65nm designs [LAFI, 2011] since the minimal TSV diameter trends are close to 1um [ITRS, 2011].

Figure 6.13 - TSV characteristics.

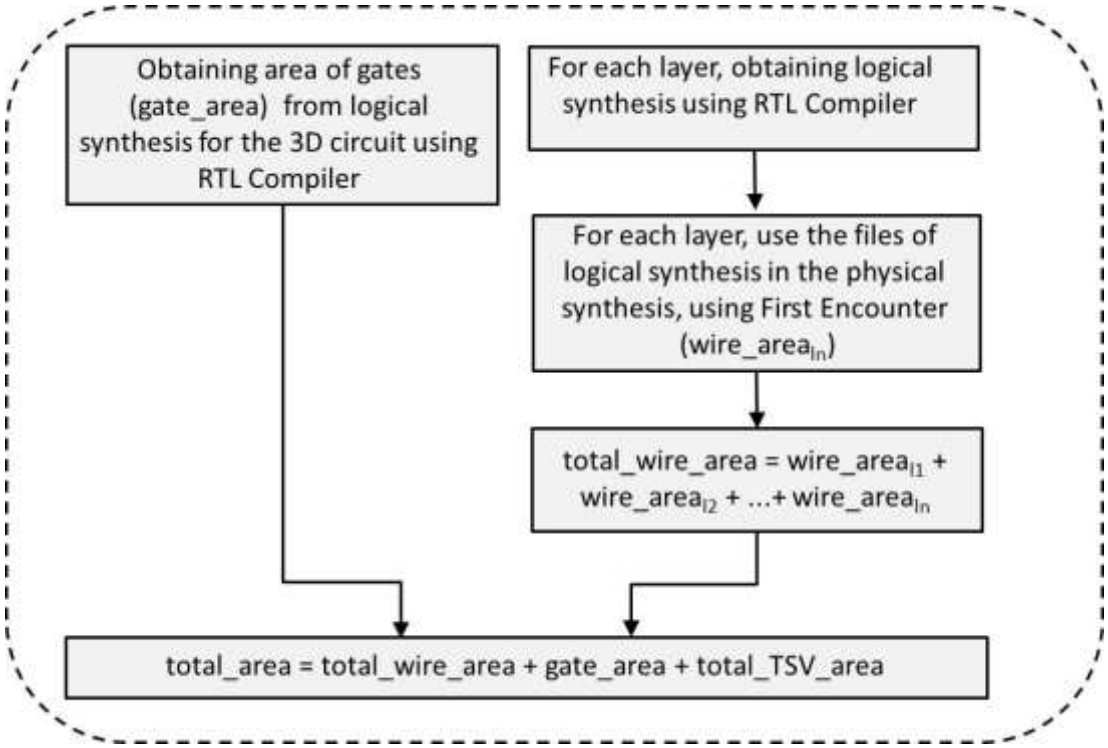


Source: elaborated by the author.

For the power results estimation, a similar analysis was considered, but in this case, as the First Encounter tool does not present the wire power separated of the total power, this value need to be discounted of the gate power for each layer. After having the total wire power, this value can be summed to the gate power of the 3D circuit. As mentioned previously, the nine metal layers were defined for routings in the First Encounter synthesis. This process is required to extract the wire parasitic information (capacitances and resistances) to obtain accurate power results.

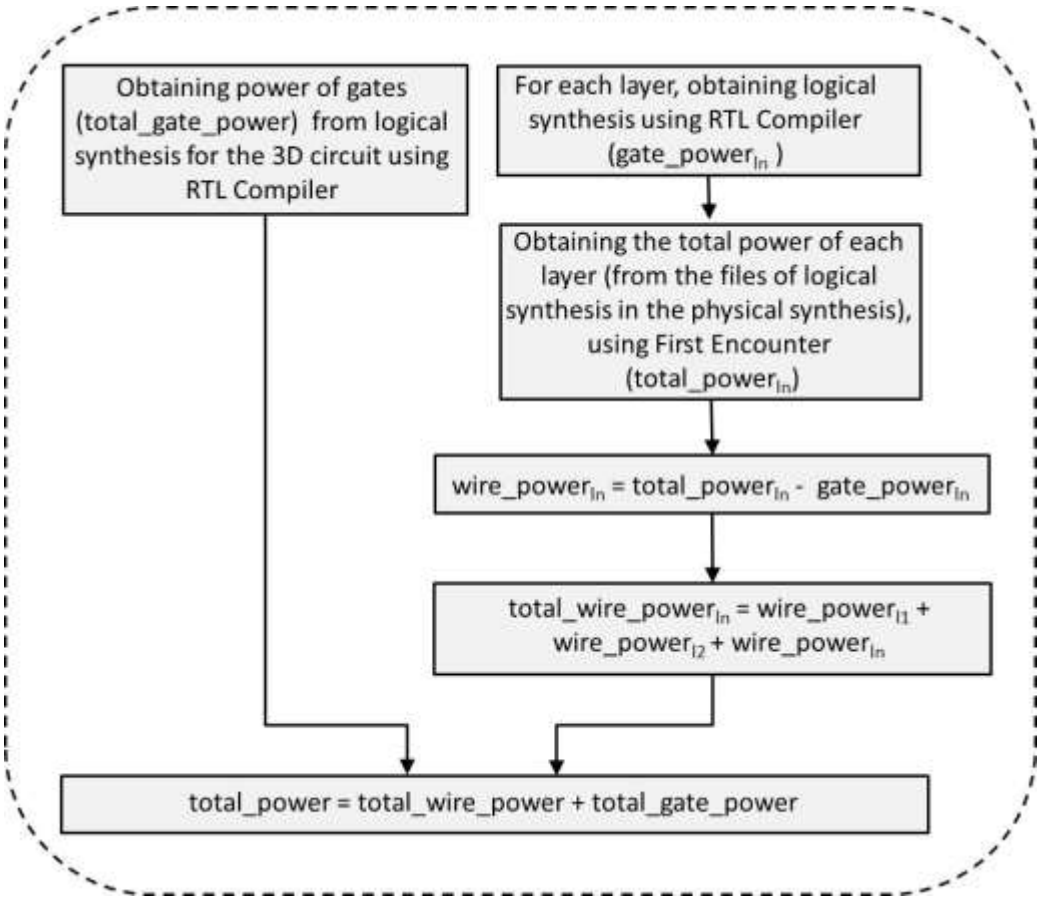
For this results, the routers present 16-bit link wide, 4-flit deep buffers, and operating frequency equal to 1GHz for 65nm process technology. These results consider a circular FIFO in the input channels of the routers, a XY routing algorithm and a wormhole switching.

Figure 6.14 - Flow chart to estimate the area for the 3D designs.



Source: elaborated by the author.

Figure 6.15 - Flow chart to estimate the power for the 3D designs.



Source: elaborated by the author.

The first experiments demonstrate the scalability of the proposal. As it is not possible to find in industry or in the literature heterogeneous applications composed of hundreds of cores, cores with 1mm² area were defined, as considered previously in other analyses. For these experiments, as in the chapter 3, two sizes for HiCIT clusters were considered: 4 and 8.

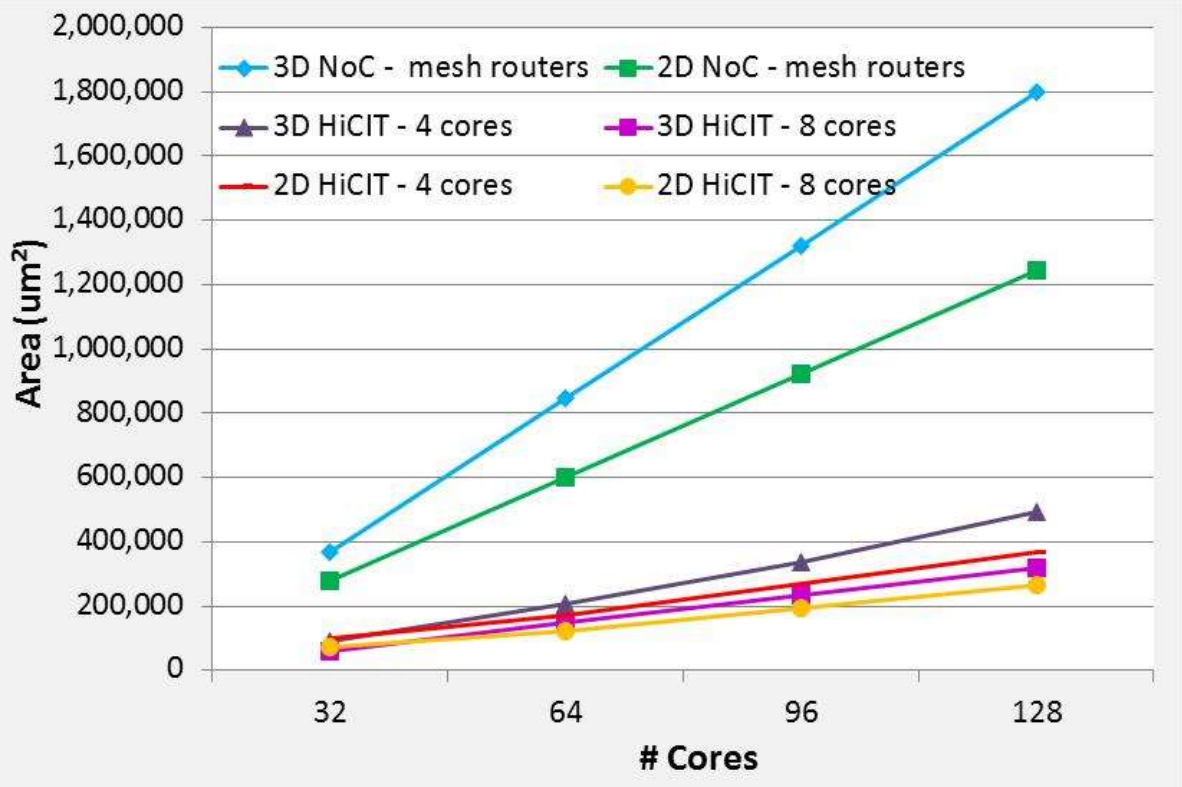
Table 6.1 presents the NoC sizes for each topology and the number of cores defined to provide the architectural comparisons. In order to have more results in terms of number of cores, it was not possible to define the same increment in the NoC sizes, for example an increment of the same number of rows and columns for each topology. In this case, the NoC sizes were adapted in order to have a topology with the same number of cores. Because of this, small variations in the curves are accepted, since the number of edge routers presenting fewer ports is different for each topology.

Table 6.1 - Topology, NoC size and number of cores defined in the experiments.

Topology	NoC size	Number of cores
3D mesh NoC	4x4x2 NoC	32
	4x4x4 NoC	64
	4x6x4 NoC	96
	4x8x4 NoC	128
	4x8x5 NoC	160
2D mesh NoC	8x4 NoC	32
	8x8 NoC	64
	8x12 NoC	96
	8x16 NoC	128
	10x16 NoC	160
3D HiCIT - 4 cores	2x2x2 HiCIT (cluster = 4 cores)	32
	2x2x4 HiCIT (cluster = 4 cores)	64
	2x3x4 HiCIT (cluster = 4 cores)	96
	2x4x4 HiCIT (cluster = 4 cores)	128
	2x4x5 HiCIT (cluster = 4 cores)	160
3D HiCIT - 8 cores	1x2x2 HiCIT (cluster = 8 cores)	32
	2x2x2 HiCIT (cluster = 8 cores)	64
	2x2x3 HiCIT (cluster = 8 cores)	96
	2x2x4 HiCIT (cluster = 8 cores)	128
	2x2x5 HiCIT (cluster = 8 cores)	160
2D HiCIT - 4 cores	2x4 HiCIT (cluster = 4 cores)	32
	4x4 HiCIT (cluster = 4 cores)	64
	4x6 HiCIT (cluster = 4 cores)	96
	4x8 HiCIT (cluster = 4 cores)	128
	5x8 HiCIT (cluster = 4 cores)	160
2D HiCIT - 8 cores	2x2 HiCIT (cluster = 8 cores)	32
	4x2 HiCIT (cluster = 8 cores)	64
	4x3 HiCIT (cluster = 8 cores)	96
	4x4 HiCIT (cluster = 8 cores)	128
	5x4 HiCIT (cluster = 8 cores)	160

The area results are presented in Figure 6.16. From these results the large area occupied by the mesh topology when compared to HiCIT is easily evidenced. The area reductions are larger than 80% when compared 3D-mesh to 3D-HiCIT with clusters composed of 8 cores and larger than 70% for cluster composed of 4 cores. As in the 2D HiCIT topology presented in Chapter 3, clusters with 8 cores present smaller area than clusters composed of 4 cores. 3D designs present larger area than 2D because the 3D routers present 2 more ports, which require more buffers, large crossbar and arbiters.

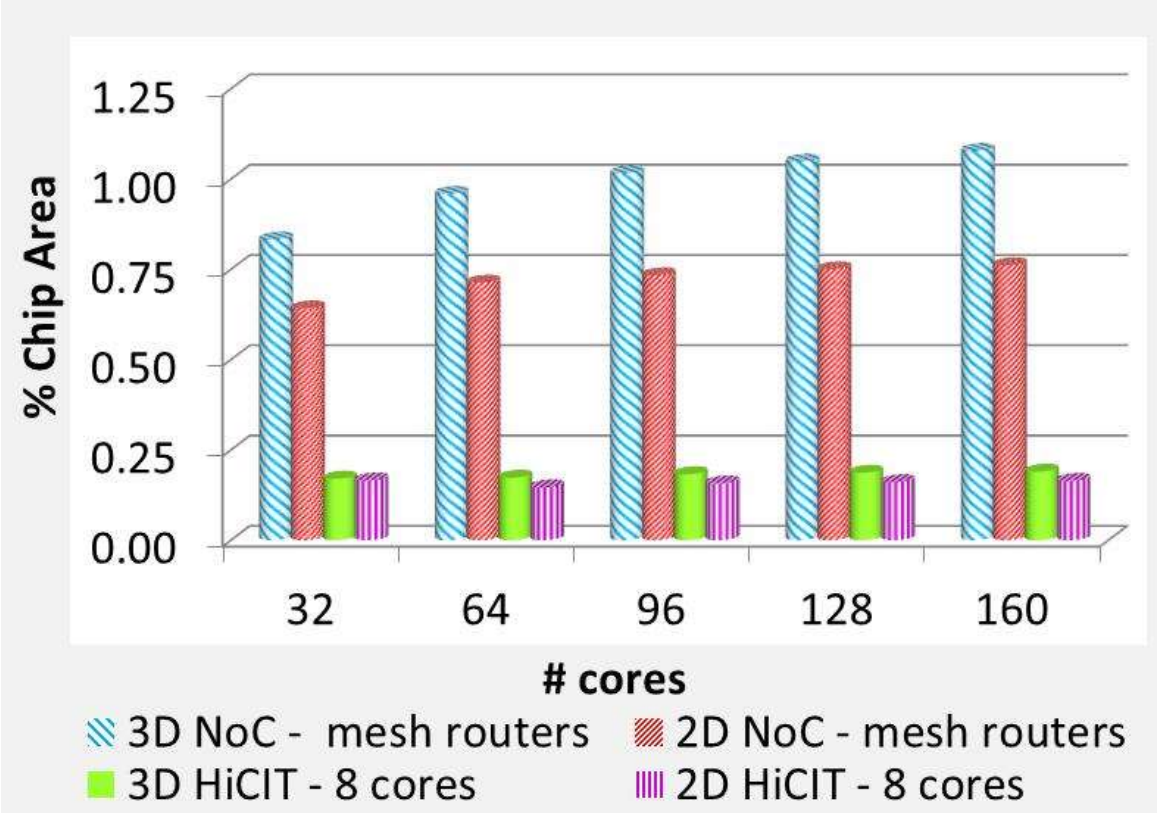
Figure 6.16 - Area estimation comparing HiCIT and mesh topologies.



Source: elaborated by the author.

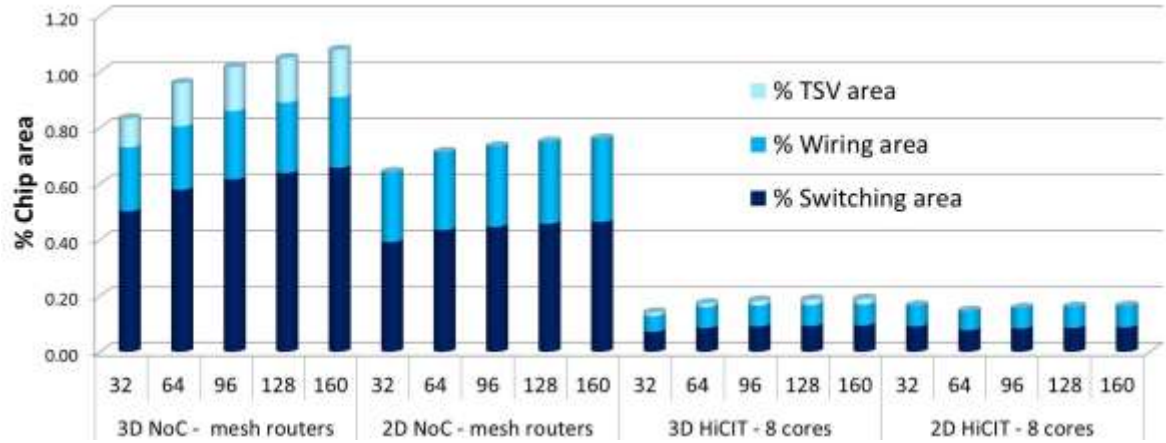
Figure 6.17 illustrates the percentage of chip area for NoCs and HiCITs for 2D and 3D topologies. As the cores have 1mm² area, which is considered a large area, the percentage of chip area was very small and smaller still for HiCIT architecture. In Figure 6.18, the same results as presented in figure 6.18 were shown but now analyzing the percentage related to wiring, switching and TSV areas. From these experiments it is important to note the high costs related to the wires and TSVs in a NoC. For mesh NoCs, the wires occupy around 40% of the NoC area, and for HiCITs, this percentage is around 50%, since crossbars present more wires. For these experiments, the TSVs do not have a large impact on the total NoC area. One reason for this is because the TSV links in this work were set to 18 bits for each vertical port (16-bit data and 2-bit control to define the flit type), which is considered small if compared to other analyses ((LAFI, 2011) used 184 links in the analysis and (XU, 2010), 128 links).

Figure 6.17 - Percentage of chip area for each NoC topology.



Source: elaborated by the author.

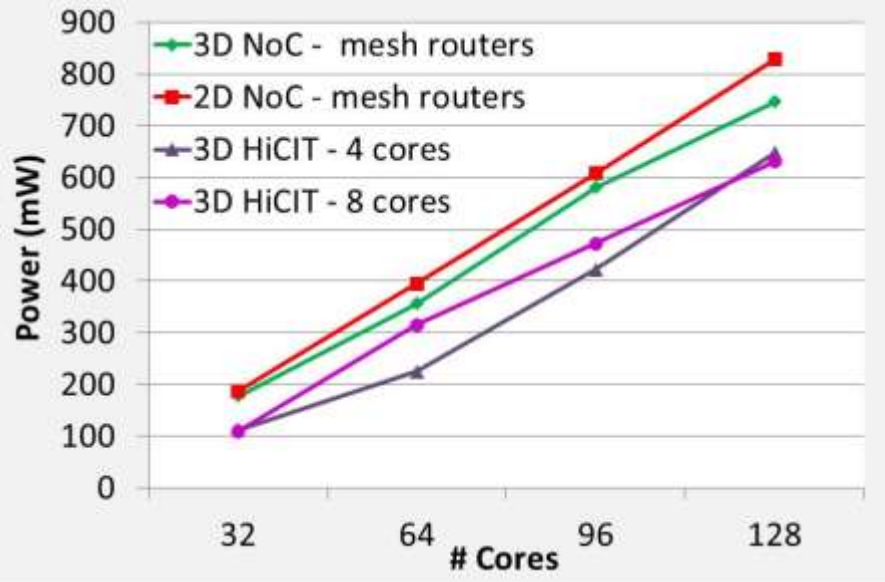
Figure 6.18 - Percentage of chip area for wiring, switching and TSVs.



Source: elaborated by the author.

Power results are presented in Figure 6.19. Again a significant reduction in the power consumption for HiCIT when compared to a mesh topology is obtained. This reduction reaches up to 38% for HiCIT composed of 8 cores. These power results consider the standard switching activity of the tool.

Figure 6.19 - Power estimation comparing 3D HiCIT and mesh topologies.



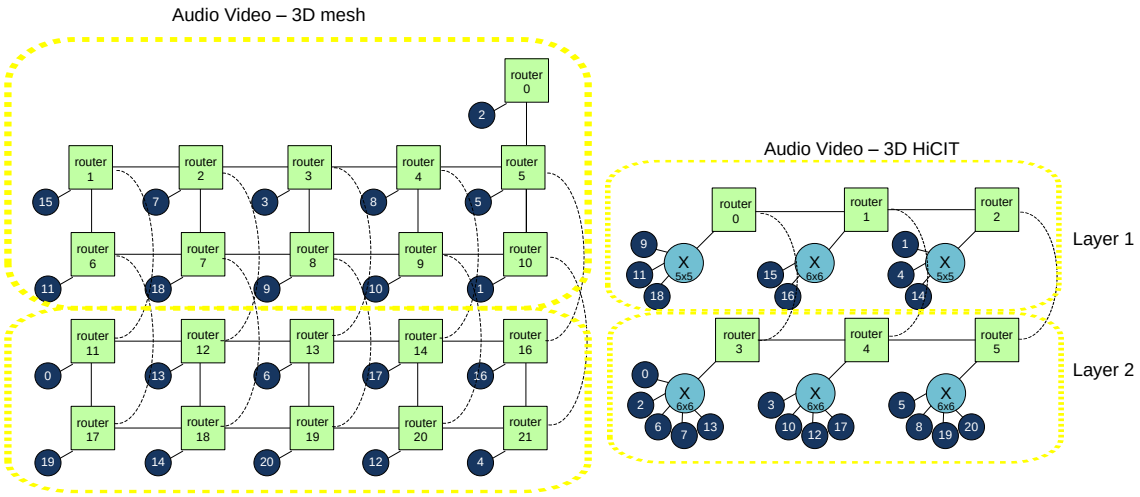
Source: elaborated by the author.

Besides the results obtained when the number of cores increases, the results for two benchmarks were also analyzed: D26 and Audio-Video (AV). As TVOPD and TMPEG4 used in the other experiments are benchmarks with cores triplicate, and in this case, with few communications between each decoder, stacking each decoder in one layer, little communication would occur between the layers. NCS already used in the other experiments, presents few cores, which complicates 3D-HiCIT analyses. Then, D26 and AV were considered in these analyses. D26 and AV benchmarks were mapped according to (TINO, 2011). This mapping tool uses a multi-objective tabu search.

The mapping for the 3D mesh and 3D-HiCIT for D26 are illustrated in Figure 6.20(a) and Figure 6.20(b), respectively and the mapping for 3D mesh and 3D-HiCIT for AV benchmark are illustrated in Figure 6.21(a) and Figure 6.21(b), respectively. The core identification for AV and D26 benchmarks were presented in Table 5.4 and Table 5.5, respectively.

Figure 6.20 - Mapping for AV benchmark for (a) 3D-mesh and (b) 3D-HiCIT topologies.

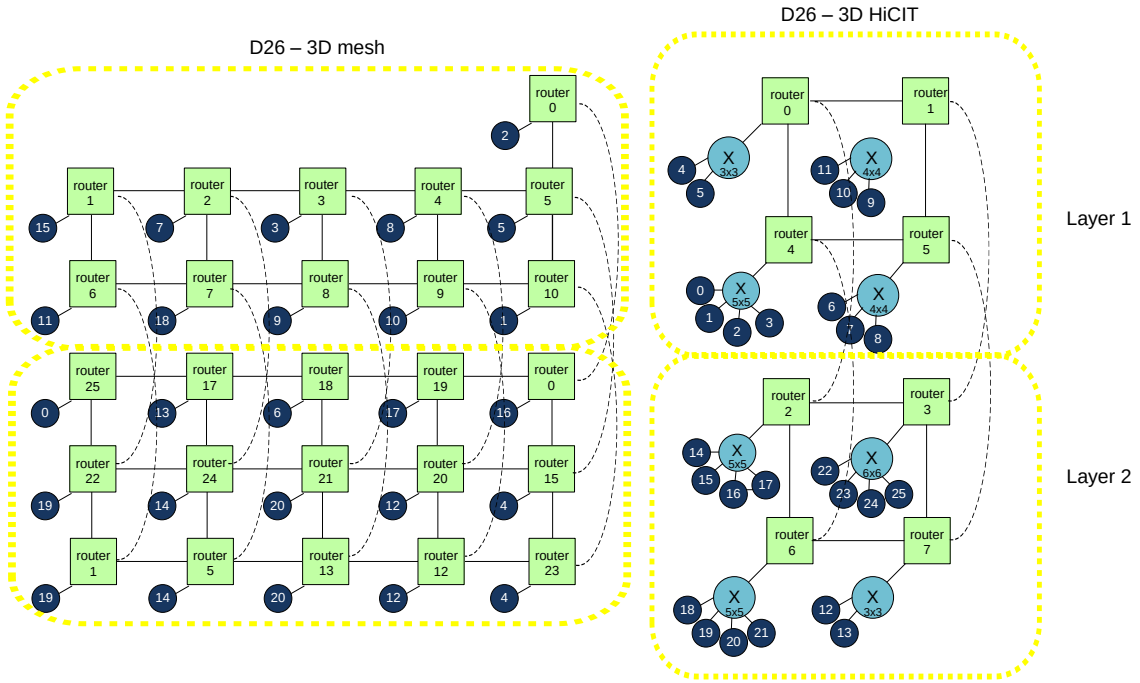
(a) (b)



Source: elaborated by the author.

Figure 6.21 - Mapping for D26 benchmark for (a) 3D mesh and (b) 3D-HiCIT topologies.

(a) (b)



Source: elaborated by the author.

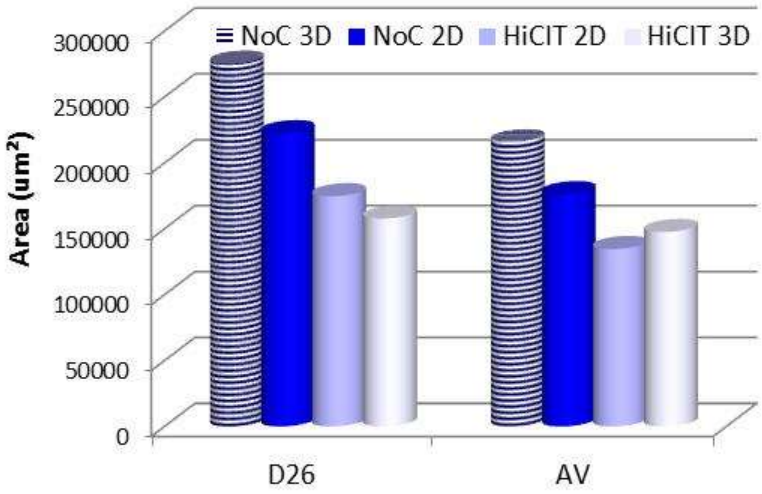
D26 core areas were extracted from (KUMAR, 2011) and the AV core areas were estimated from similar cores used in TSV and NCS, since accurate area information for the cores of this benchmark was not found in the literature. Table 6.2 presents the core areas used for AV benchmark.

Table 6.2 - Physical information of AV cores.

Core	area (mm^2)
DSP1 – DSP8	0.2276000
CMEM1 – CMEM2	0.976358
MEM1- MEM2	0.976358
CPU	0.979196
ASIC1	0.992299
ASIC2M	0.992299
ASIC2S	0.992299
ASIC3S	0.992299
ASIC3M	0.992299
ASIC4S	0.992299
ASIC4M	0.992299

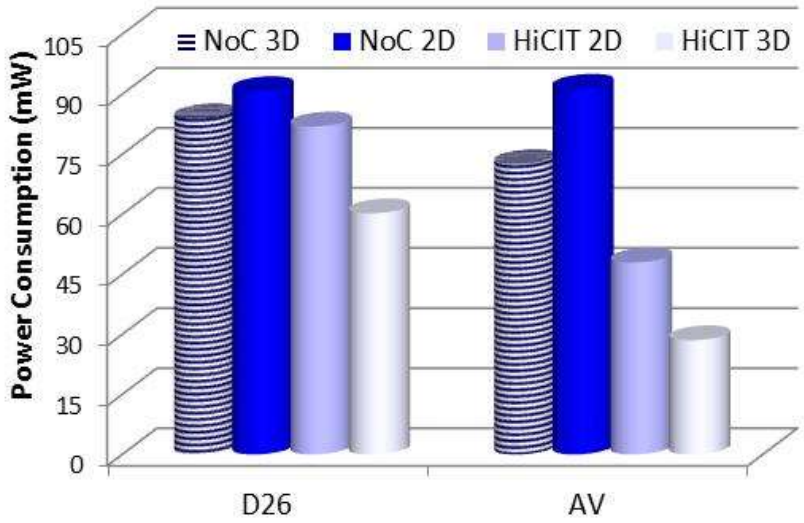
Area and power results for these benchmarks are presented in Figure 6.22 and Figure 6.23, respectively. Following the previous results, for these two benchmarks the advantages in the use of HiCIT for 3DICs are also evidenced. As can be inferred from these results, 3D-HiCIT presents an even smaller power consumption and area than HiCIT for 2D designs, combining the advantages of the 3D integration with the hierarchical proposal. According to mappings of Figure 6.20 and Figure 6.21, one core was maintained isolated in the first row, since specifically for these applications, as it uses XYZ routing algorithm, problems in the message routing do not occur. However, if fully interconnected mesh is used in this comparison, further reductions are reached.

Figure 6.22 - Area results for D26 and AV benchmarks comparing mesh and HiCIT topologies.



Source: elaborated by the author.

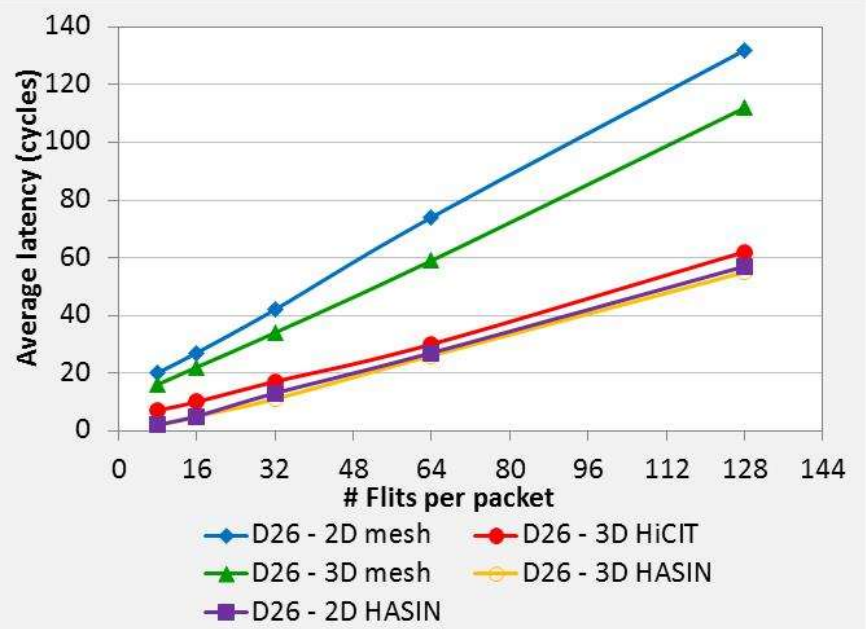
Figure 6.23 - Power Consumption for D26 and AV benchmarks comparing mesh and HiCIT topologies.



Source: elaborated by the author.

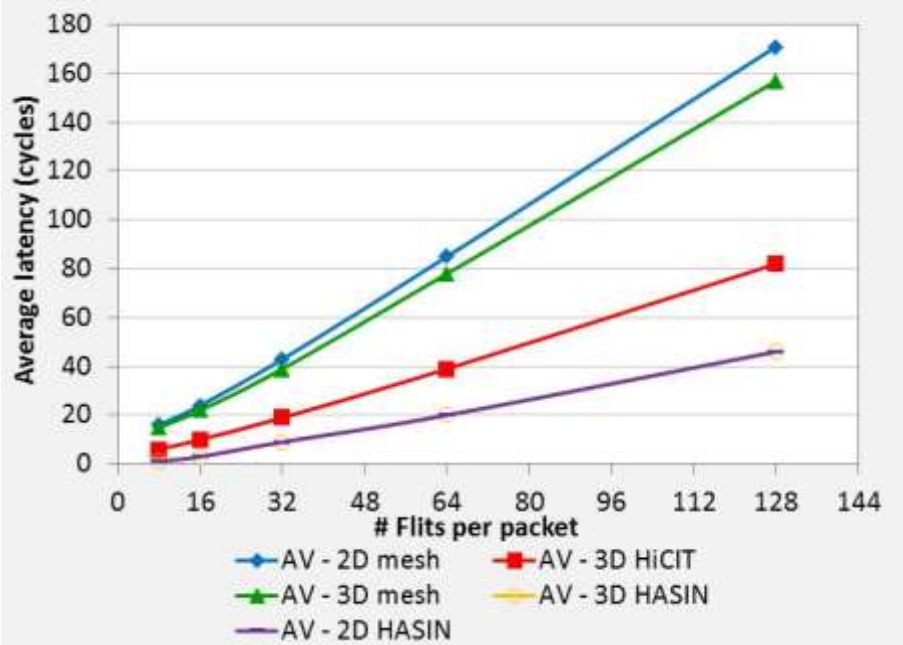
Figure 6.24 and Figure 6.25 present the average latency for D26 and AV benchmarks, respectively. For this experiment, the same Java simulator described Chapter 5 was used to simulate the applications. As can be verified, the HiCIT proposal is also scalable for 3D topologies, reducing the latency up to 60% for packets composed of 8 flits and around 45% for packets composed of 128 flits. The results for HASIN proposal was also verified for these benchmarks. However, as 3D designs already reduce the latency and HiCIT further decrease this value, for these benchmarks that are composed of few cores, for these new context is not advantageous the use one more technique, as adaptability. As HASIN increases the power consumption when compared to HiCIT and the advantages using this topology for these benchmarks are insignificant, the best topology among the compared ones for these benchmarks is the 3D-HiCIT. 3D HiCIT allows reducing power, area and average latency when compared to a mesh topology.

Figure 6.24 - Average latency comparisons considering mesh and HiCIT topologies for D26 benchmark.



Source: elaborated by the author.

Figure 6.25 - Average latency comparisons considering mesh and HiCIT topologies for D26 benchmark.



Source: elaborated by the author.

6.4 Considerations

In this chapter, important issues related to the 3D integrations were considered, such as the advantages and limitations of these designs. After this discussion, relevant aspects of 3D interconnection were highlighted which point to a hierarchical solution. In this case the

HiCIT topology was analyzed for 3DICs, since this topology is well suited to these designs. Results indicated a reduction in power consumption and area when compared to the mesh topology. Latency results also demonstrated the advantages of this solution when compared with other topologies.

7 CONCLUSION AND FUTURE WORKS

During the last years, it has been possible to see a significant evolution in the many and multi core systems through the aggressive scaling of CMOS technology in the nanoscale era. In this direction, MPSoCs have presented a continual increase in the complexity, and so an interconnection for these systems has faced severe constraints. The first NoC proposals were believed to meet the requirements of these systems. However, foreseeing systems composed of hundreds of cores, old NoC architectures demonstrated performance bottlenecks. Due to many reasons, the NoCs needed to adopt innovative mechanisms. The use of an intelligent interconnection device able to be self-adaptable becomes essential in networks-on-chip. Hierarchy in NoCs is another important feature in order to provide the system bandwidth requirements and efficient power-aware interconnection, since it is able to explore the communication locality.

Along with these NoC alternatives, 3D integration is perceived as a solution to scale the performance of electronic devices beyond Moore's law. 3D integrations have emerged as a solution to obtain smaller circuit footprint, shorter vertical links, latency and a reduction in power consumption when compared to old solutions. This new integration possibility also has other advantages such as noise isolation and the possibility of integrating mixed technologies, offering a greater opportunity to diversify the functionality of electronic devices. In order to provide all these advantages in a single interconnection device, it is evident that a set of techniques needs to be adopted.

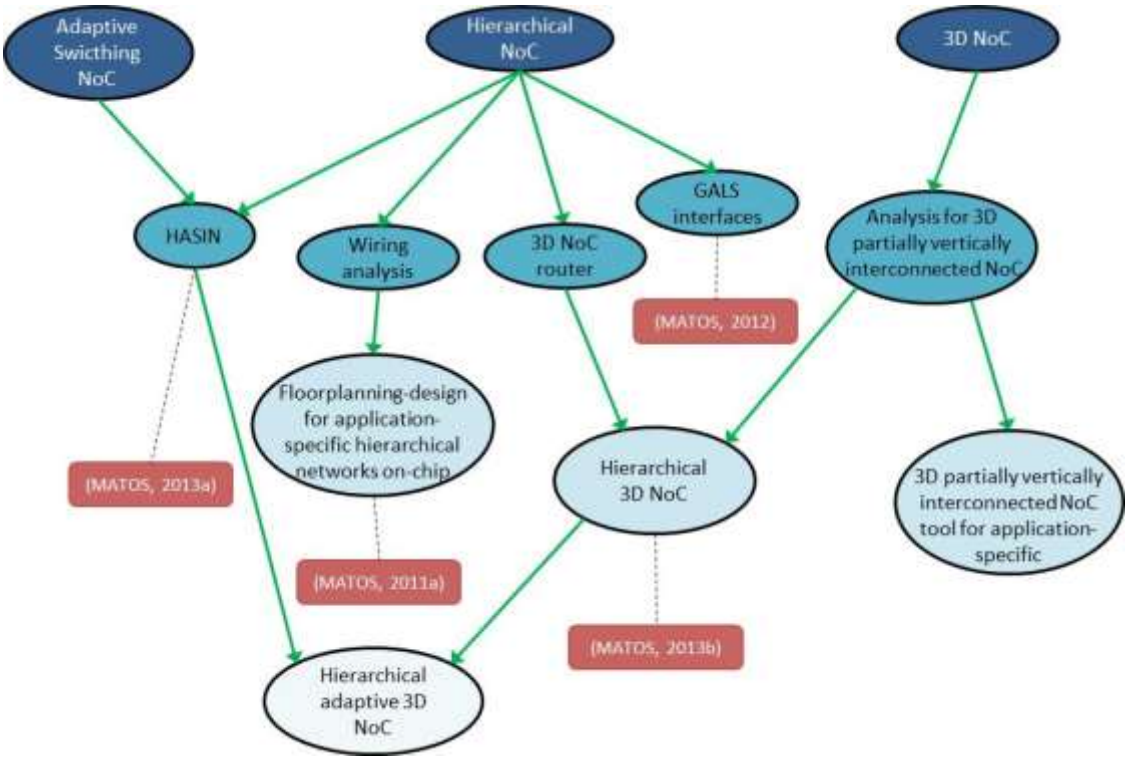
In the next section, the major contributions of this thesis and how this work was originally conceived and conducted, presenting also the work collaborations during this period is presented. Finally, section 7.2 lists some future directions and open issues to continue the research and investigations in the context of NoCs.

7.1 Major contributions

This work joined a set of interesting techniques that applied in a system can obtain an efficient interconnection network solution. The planned strategies comprehend: hierarchy, adaptability and 3D interconnection.

Figure 7.1 presents the flow diagram of how the work of this thesis was conducted. The first proposals of this figure illustrates the research lines initially planned. The topology is one of the most important features of a NoC since the communication performance and NoC power consumption are totally dependent on this. However, the choice of the best topology candidate to interconnect a system is a challenge, since the limitation of each solution needs to be thoroughly evaluated. Following the requirements of complex and heterogeneous MPSoCs expected for the next generation, a hierarchical NoC topology seemed the most appropriated interconnection. In this way, the possibility of using crossbars in a hierarchical manner was analyzed. For this and in order to understand the scalability of this strategy, some experiments were verified such as the impact of long wires using this interconnection. With this analysis the possibility of using a topology composing clusters with mesh routers and crossbars was confirmed. In such a case, the clusters can be heterogeneous and the sizes of the crossbars are defined according to the application requirements. The scalability of this proposal is reached in response to the cluster structure.

Figure 7.1 - Strategies approached in this work.



Source: elaborated by the author.

After the initial studies related to interconnection hierarchy, a tool to obtain the mapping for an application-specific interconnection by the proposed hierarchical NoC was developed in collaboration with the group Dipartimento di Elettronica e Informazione at Politecnico di Milano (MATOS, 2011a).

One important feature related to hierarchical topologies is the possibility of using GALS in the system. In this case, the clusters can compose islands operating at a specific frequency, which allows a power-efficient system. One approach using GALS interfaces was applied to the hierarchical proposal using bridges between the router and the crossbar (MATOS, 2012).

One of the goals in the definition of the hierarchical proposal was obtaining a scalable topology able to be applied in the 3D design context. As well as this, other strategies were analyzed in parallel such as adaptive mechanisms for NoCs. Adaptability in NoCs will be required since the systems are heterogeneous under different aspects. Besides, modern systems are capable of having tasks updated, impacting the communication bandwidths. For this reason, an adaptive NoC must be able to achieve high performance in all these situations. In order to cope with this

requirement, innovative adaptive switching was researched and developed. Thus, the proposed strategy takes into account the floorplan information to define the switching mode.

Due to the fact that hierarchical NoCs are usually defined to be application-specific, the network needed to present an alternative whenever the communication pattern suffered alterations (MATOS, 2013a). Because of this, adaptive switching was integrated in the final NoC solution. Thus, the proposed architecture is able to deal with different traffic situations in the network, ensuring high performance with minimal costs and power consumption.

Lastly, with so many factors pointing toward the use of 3D integration to continue the MPSoC evolution, some relative studies were initiated in this thesis. During the collaboration with the CEA/Leti laboratory, some studies regarding the advantages and limitations of the 3D NoCs were cleared, since this group works extensively on 3D integration research, having already implemented some 3D chip prototypes. 3D integration presents some important advantages, such as power reduction, possibility to integrate a specific technology in each layer and the noise isolations among analog/ RF and digital circuits. However, 3DIC technologies also bring new challenges. From these studies, the efficiency in the use of the HiCIT topology in this new context was proved. As 3D NoCs using TSVs is relatively new, there are several aspects that need further investigation. As such, there are important issues where the research of this thesis can continue, as will be better detailed in the next section.

Summarizing, below the major contributions of this work are listed:

- **A hierarchical proposal able to drastically reduce the latency and power consumption:** Hierarchy is a simple strategy to achieve high performance for an application, since it is possible to provide a faster communication among the high bandwidth cores in a cluster disposition, exploring the communication locality. In the proposed solution, the routers of a mesh topology are replaced by crossbars. As crossbars are simpler components than routers, without buffers and extra controls, this strategy allows also a large reduction in the power consumption.
- **An innovative adaptive switching strategy to increase the performance from floorplan information:** The most interesting aspect

of this strategy is the possibility of defining the NoC switching mode based on the wire length. In this manner, this strategy meets low latency combined with high frequency. However, it consumes more power than a conventional packet switching NoC.

- **The possibility of integrating the adaptive strategy in the hierarchical topology in order to ensure high performance even when the system traffic pattern is altered:** The integration of the hierarchical solution with the adaptive strategy presents several advantages. As the hierarchical proposal is initially defined to be application-specific, the use of an adaptive strategy adds some flexibility to the interconnection. Besides, as the hierarchical NoC consumes lower power than mesh topologies, the addition of resources to provide the adaptability leads to an increment in the power consumption. Then, in comparison with the mesh topology, besides increasing the performance, this solution reduces the power consumption and allows variations in the core communication.
- **A thorough study concerning the 3D topologies that point out the use of hierarchical NoC:** 3D integration presents many advantages, as already mentioned. Nevertheless, like any new technology it also presents some challenges. One limitation is related to the difficulty to insert a large number of TSVs, since they require large pads, substantially increasing the area. Using the proposed hierarchical NoC is a very interesting solution for 3D integration and this was proved by the results.

In order to prove that this work has complied with the challenges initially raised, it is interesting to look again at the table presented in the introduction, but now restated in Table 7.1 with the publications obtained for each solution during the PhD course, evidencing the importance of each proposal for the research.

Table 7.1 - Solutions for each challenge and the related publications.

Challenges	Solutions	Chapters
1 Large systems increase the system latency	✓ Hierarchical interconnections ✓ Adaptive strategy ✓ 3D NoCs	(MATOS, 2012) (MATOS, 2013a) (MATOS, 2013b)
2 High performance associated with low power	✓ GALS ✓ 3D NoCs ✓ Hierarchical interconnections ✓ Adaptive strategy + hierarchical topology	(MATOS, 2012) (MATOS, 2013a) (MATOS, 2013b) (MATOS, 2011b)
3 Clock skew and jitter	✓ GALS	(MATOS, 2012)
4 Need to meet different system requirements	✓ Adaptive strategy	(MATOS, 2013a)
5 Heterogeneous Systems	✓ Hierarchical interconnections	(MATOS, 2011a) (MATOS, 2012)
6 Increase in the wire delay and wire power	✓ 3D NoCs	(MATOS, 2013b)

Results demonstrated the large advantages obtained with the proposed solutions in this work. With the integration of all techniques presented in this thesis, it is possible to obtain an NoC architecture capable of meeting different MPSoC requisites. The major gains of this proposal are possible due to the use of a very low cost hierarchical architecture, such as HiCIT solution. Thereby allowing a reduction in the power consumption and increasing the performance, since a hierarchical arrangement favors communication of cores with large bandwidths. With the use of adaptive strategy combined with the hierarchical solution, it became possible to obtain further advantages, since the increment in power consumption and area when compared to HiCIT are small but when compared to a conventional mesh, the reduction in the total power consumption and area are very impressing (both around 70%). Besides, the benefits in performance are much larger when compared to a mesh NoC. HiCIT is also a

good alternative for 3D designs. 3D-HiCIT presents an even smaller area and power consumption when compared to HiCIT for 2D designs. Combining the advantages of the 3D integration with the hierarchical proposal is a very interesting solution for this new context of design integration.

7.2 Open issues and future directions

Due to the increase in the complexity of the MPSoCs, with a pressure and demand for a communication infrastructure that is able to provide the required bandwidth, energy efficiency, scalability and performance stability under different traffic conditions, the NoCs has received special attention and a solution that satisfies all these conditions is constantly being pursued. Because of this, new strategies have emerged and some of them present challenges that still require much research. Here are some research areas to be continued after this work.

- **3D hierarchical NoC focusing on QoS and application deadline:** 3D designs can include multi-functionality, increase performance and reduced power when compared to the 2D integration. All these benefits make 3D stacking a promising solution to overcome present limitations of aggressive CMOS technologies. In this thesis, some initial studies were presented for a 3D hierarchical NoC proposal. Moving forward from this point, the research purpose is to continue the studies related to 3D topologies. The gains with 3D-HiCIT are clear, however the next step to be investigated in this proposal is to analyze the QoS (quality-of-service) of the proposed solution, in terms of the application deadline of this solution. One feature that needs to be investigated is to measure if, when the system runs a different application, all communication deadlines are served. It is possible that some channels are not meeting the required bandwidth, while other channels are getting much more bandwidth. Verifying the average results, the performance is improved with the proposed strategies, but it is not known if all application demands were met. Providing quality-of-service guarantees for concurrent tasks in many-core architectures is essential especially for real-time applications (OUYANG, 2010).
- **Exploring other adaptive techniques for 3D designs**: it is known that the use of many strategies integrated in a communication infrastructure obtains many advantages to execute heterogeneous and different applications, that present

varied traffic conditions. Exploring adaptive techniques for 3D NoC is a recent research topic with few related studies. In this manner, this is an important topic where the research can be continued. Understanding the need for 3D stacked circuits, the allocation in the chip of each layer according to the technology and the real traffic among the layers are some situations not totally defined. Continued study in order to better understand the needs and challenges of 3D integration is an important task to propose other and maybe more appropriate adaptive techniques.

- **Exploring the mapping and architecture of 3D hierarchical NoCs taking into account communication performance, power consumption and thermal aspects**: obtaining the ideal 3D mapping for an application taking into account all relevant aspects of 3D designs is not an easy task. Some works present 3D mapping solutions taking only some aspects into consideration (RAHMANI, 2011b), (CHAO, 2010), (SEICULESCU, 2009). With the 3D integration, another challenge needs to be observed: the chip temperature. As more layers are stacked vertically, larger is the power density (GRANGE, 2011). High temperatures may limit the viability of the 3D same-die stacking approach by reducing the operating frequencies of vertically-stacked cores and degrading chip reliability. Potential solutions for this critical issue may be found in an optimal mapping and NoC architecture. This is also an important topic that is being evidenced in the research and that need to be considered in proposals.
- **Exploring the possibility of obtaining interconnection infrastructures that integrate mixed technologies:** this is an interesting research motivation, since new technologies have been proposed and the use of a network that combines CMOS with other technologies can be a solution to accelerate the communications and/or reduce the power consumption in large MPSoCs (GU, 2009), (MO, 2010), (MORRIS, 2012). Optical NoCs promise significant bandwidth and power advantages (MO, 2010). As the power dissipation of metallic interconnections in manycore architecture has presented values close to the power dissipation of the gates, alternative power-efficient technology solutions have been developed for future integrated circuits and it is a very interesting way to follow the research.

REFERENCES

AKBARI, S. et al., AFRA: A low cost high performance reliable routing for 3D mesh NoCs, **Proceedings**... Design, Automation & Test in Europe (DATE), San Jose: EDA Consortium, p. 332 - 337, Mar. 2012.

AMBA Specification (rev2.0) – Available at:

http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.set.amba/index.html, Access in: 20 Dec. 2013.

ARM, Available at http://www.arm.com/products/processors/classic/index.php, Access in: 20 Dec. 2013.

ARTERIS, Available at http://www.arteris.com/, Access in: 20 Dec. 2013.

ATIENZA, D. et al., Network-on-Chip Design and Synthesis Outlook, **Integration, the VLSI Journal**, Amsterdam: Elsevier Science Publishers B. V., Vol. 41, Issue 3, p. 340 - 359, May 2008.

BANERJEE, K. et al., 3-D ICs: A Novel Chip Design for Improving Deep-Submicrometer Interconnect Performance and Systems-on-chip Integration, **Proceedings**…IEEE, [S.l.:s.n.], Vol. 89, no. 5, p. 602 - 633, May 2001.

BANERJEE, N. et al., Novel Low-Overhead Operand Isolation Techniques for Low-Power Datapath Synthesis, **IEEE Transaction on Very Large Integration (VLSI) Systems**, Piscataway: IEEE Educational Activities Department, vol. 14, Issue 9, p. 1034 - 1039, Sep. 2006.

BENINI L., DE MICHELI G. Powering Networks on Chips: Energy-Efficient and Reliable Interconnect Design for SoCs. **Proceedings**… International Symposium on Systems Synthesis, New York: ACM, p. 33 - 38, 2001.

BENINI, L., DE MICHELI G. Networks on Chips: a New SoC Paradigm, **IEEE Computer**, Los Alamitos: IEEE Computer Society Press, Vol. 35, Issue 1, p. 70 - 78, Jan. 2002.

BEIGNÉ, E.; VIVET, P. Design of On-chip and Off-chip Interfaces for a GALS NoC Architecture, **Proceedings**… IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC), Washington: IEEE Computer Society, p. 172 - 183, 2006.

BERTOZZI, D.; BENINI, L., Xpipes: A Network-on-chip Architecture for Gigascale Systems-on-Chip, **Circuits and Systems Magazine**, [S.l.]: IEEE, vol. 4, no. 2, p.18 - 31, Sep. 2004.

BERTOZZI, D. et al. NoC Synthesis Flow for Customized Domain Specific Multiprocessor Systems-on-Chip, **IEEE Transaction on Parallel and Distributed System**, Piscataway: IEEE Press, p. 113 - 129, Feb. 2005.

BJERREGAARD, T., MAHADEVAN, S., A Survey of Research and Practices of Network-on-Chip. **ACM Computing Surveys**, New York: ACM, Vol. 38, Issue 1, 2006.

BLENDICS, available at http://www.blendics.com/ Available in 20 Dec. 2013.

BOBDA, C. et al., DyNoC: A dynamic infrastructure for communication in dynamically reconfigurable devices, **Proceedings**… International Conference on Field Programmable Logic and Applications, [S.l.:s.n.], p.153 - 158, Aug. 2005.

BOGATIN, E., Chapter 5: The Physical Basis of Capacitance, **Signal and Power Integrity – Simplified**, 2nd Edition, Boston: Prentice Hall, 2009.

BOLOTIN, E. et al. QNoC: QoS Architecture and Design Process for Network on Chip. **Journal of Systems Architecture: the EUROMICRO Journal** - Special issue: Networks on chip, New York: Elsevier North-Holland, Vol. 50, Issue 2-3, p. 150 - 128, Feb. 2004.

BOLOTIN, E. et al. Routing Table Minimization for Irregular mesh NoCs**. Proceedings**... Conference on Design, Automation and Test in Europe (DATE), San Jose: EDA Consortium, p. 942 - 947, 2007.

BOURDUAS, S.; ZILIC, Z., A Hybrid Ring/Mesh Interconnect for Network-on-Chip Using Hierarchical Rings for Global Routing, **Proceedings**… NOCS '07 Proceedings of the First International Symposium on Networks-on-Chip, Washington: IEEE Computer Society, p.195 - 204, May 2007.

CHAN, J.; PARAMESWARAN S. NoCOUT: NoC Topology Generation with Mixed Packet-Switched and Point-to-Point Networks. **Proceedings**… Asia and South Pacific Design Automation Conference (ASP-DAC), Los Alamitos: IEEE Computer Society Press, p. 265 – 270, 2008.

CHANG, K.; SHEN, J.; CHEN, T., Evaluation and Design Trade-offs Between Circuit-Switched and Packet-Switched NOCs for Application-Specific SOCs. **Proceedings**… 43rd annual Design Automation Conference (DAC). New York: ACM, p. 143 - 148, 2006.

CHAO, C. et al., Traffic and Thermal Aware Run Time Thermal Management Scheme for 3D NoC Systems, **Proceedings**… 2010 Fourth ACM/IEEE International Symposium on Networks-on-Chip (NoCS), Washington: IEEE Computer Society, p. 223 - 230, 2010.

CHEN, X.; PEH, L., Leakage Power Modeling and Optimization in Interconnection Networks, Low Power Electronics and Design, **Proceedings**… International Symposium on Low Power Electronics and Design (ISLPED), New York: ACM, p. 90 - 95, Aug. 2003.

CHOU, S. et al., Hierarchical Circuit-Switched NoC for Multicore Video Processing. **Microprocessors & Microsystems**, Amsterdam: Elsevier Science Publishers B. V., Vol. 35, no 2, p. 182 - 199, 2011.

CORREALE, A., Overview of the power minimization techniques employed in the IBM PowerPC 4xx embedded controllers. **Proceedings**… 1995 International Symposium on Low power Design (ISLPED). New York: ACM, p. 75 - 80, 1995.

DALLY, W.; TOWLES, B. Route Packets, Not Wires: On-Chip Interconnection Networks. **Proceedings**… Design Automation Conference (DAC), New York: ACM, p. 684 - 689, 2001.

DALLY, W.; TOWLES, B. **Principles and Practices of Interconnection Networks**. San Francisco, Morgan Kaufmann Publishers Inc., San Francisco, 550 p., 2003.

DAS, R., et al., Design and Evaluation of a Hierarchical On-Chip Interconnect for Next-Generation CMPs, **Proceedings**… IEEE 15th International Symposium High Performance Computer Architecture (HPCA), [S.l]: IEEE, p. 175 – 186, Feb. 2009.

DANESHTALAB, M.; et al., CMIT — A Novel Cluster-Based Topology for 3D Stacked Architectures, **Proceedings**… IEEE International 3D Systems Integration Conference (3DIC), [S.l]: IEEE, p. 1 - 5, Nov. 2010.

DAVIS, W. R. et al., Application Exploration for 3D Integrated Circuits: TCAM, FIFO and FFT Case Studies, **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, Piscataway: IEEE Educational Activities Department, Vol. 17, Issue 4, p. 496 - 506, Apr. 2009.

DEB, K. et al., A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, **IEEE Transactions on Evolutionary Computation**, Piscataway: IEEE Press, Vol. 6, Issue 2, p. 182 - 197, Apr. 2002.

DE MICHELI, G. et al., Networks on Chip: from Research to Products. **Proceedings**… Design Automation Conference (DAC), New York: ACM, p. 300 - 305, 2010.

DE PAULO, V.; ABABEI, C., 3D Network-on-Chip Architectures Using Homogeneous meshes and Heterogeneous Floorplans. **International Journal of Reconfigurable Computing**, [S.l.:s.n.], Vol. 2010, article no. 1, Jan. 2010.

DUBOIS, F., Elevator-First: A Deadlock-Free Distributed Routing Algorithm for Vertically Partially Connected 3D-NoCs, **IEEE Transactions on Computers**, Washington: IEEE Computer Society, vol. 62, no.3, p. 609 - 615, Mar. 2013.

EBRAHIMI, M. et al., Exploring partitioning methods for 3D Networks-on-Chip utilizing adaptive routing model, **Proceedings**…Fifth ACM/IEEE International Symposium on Networks-on-Chip Networks on Chip (NoCS), New York: ACM, p. 73 - 80, May 2011.

ENQUIST, P., High Density Direct Bond Interconnect (DBI) Technology for Three Dimensional Integrated Circuit Applications. **Proceedings**… Materials Research Society - Symposium Y – Enabling Technologies for 3-D Integration, [S.l.:s.n.], Vol. 970, Jan. 2006.

FARUQUE, M.; EBI, T.; HENKEL, J., ROAdNoC: Runtime Observability for an Adaptive Network on Chip Architecture, **Proceedings**…2008 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Piscataway: IEEE Press, p. 543 - 548, Nov. 2008.

FEERO, B.; PANDE P., Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation, **IEEE Transactions on Computers**, Washington: IEEE Computer Society, Vol. 58, Issue1, p. 32 - 45, Jan. 2009.

GILABERT, F. et al., Designing Regular Network-on-Chip Topologies under Technology, Architecture and Software Constraints, **Proceedings**… International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), [S.l.:s.n.], p. 681 - 687, Mar. 2009.

GOODWIN, D.; ROWEN, C.; MARTIN, G., Configurable Multi-Processor Platforms for Next Generation Embedded Systems, **Proceedings**… Asia and South Pacific Design Automation Conference (ASP-DAC), Washington: IEEE Computer Society Press, p. 744, 746, 2007.

GOOSSENS, K.; DIELISSEN, J.; RADULESCU, A., AEthereal Network on Chip: Concepts, Architectures, and Implementations, **Proceedings**… IEEE Design & Test of Computers, [S.l.:s.n.], Vol. 22, no. 5, p. 414 - 421, 2005.

GOPLEN, B.; SAPATNEKAR, S., Placement of 3D ICs with Thermal and Interlayer Via Considerations, **Proceedings**… 44th annual Design Automation Conference (DAC), New York: ACM, p. 626 - 631, Jun. 2007.

GRANGE, M., Modeling the Computational Efficiency of 2-D and 3-D Silicon Processors for Early-chip Planning, **Proceedings**… International Conference on Computer-Aided Design (ICCAD), Piscataway: IEEE Press, p. 310 - 317, Nov. 2011.

GU, H. et al., A Low-Power Low-Cost Optical Router for Optical Networks-on-Chip in Multiprocessor Systems-on-Chip, **Proceedings**… IEEE Computer Society Annual Symposium on VLSI, Washington: IEEE Computer Society, p. 19 - 24, May 2009.

GUERRIER P.; GREINER, A., A Generic Architecture for On-Chip Packet-Switched Interconnections. **Proceedings**…Design Automation and Test in Europe (DATE), New York: ACM, p. 250 – 256, 2000.

GUERRE, A. et al., Hierarchical Network-on-Chip for Embedded Many-Core Architectures, **Proceedings**… Fourth ACM/IEEE International Symposium on Networks-on-Chip (NoCS), Washington: IEEE Computer Society, p. 189 - 196, 2010.

HELD, J. KOEHL, S., Introducing the Single chip Cloud Computer Exploring the Future of Many-Core Processors, p. 1-5, 2010. Available at: http://newsroom.intel.com/servlet/JiveServlet/previewBody/1088-102-1-1165/Intel_SCC_whitepaper_4302010.pdf. Access in: 17 Apr. 2014.

HAIBO, Z.; PANDE, P.P.; GRECU, C., Performance Evaluation of Adaptive Routing Algorithms for Achieving Fault Tolerance in NoC Fabrics, **Proceedings**… Application-Specific Systems, Architectures and Processors (ASAP), [S.l.:s.n.], p. 42 - 47, Jul. 2007.

HEMANI, A. et al., Network on chip: an Architecture for Billion Transistor Era, **Proceedings**…IEEE NorChip Conference, [S.l.:s.n.], p. 1 - 8, Nov. 2000.

HO, R.; MAI, K.W.; HOROWITZ, M.A., The Future of Wires, **Proceedings**… IEEE, [S.l.:s.n.] Vol. 89, no. 4, p. 490 - 504, Apr. 2001.

HOLLSTEIN, T., HiNoC: A Hierarchical Generic Approach for on- Chip Communication, Testing and Debugging of SoCs, **Proceedings**… IFIP International Federation for Information Processing, [S.l.]:Springer, p. 34 - 54, Vol. 200, 2006.

HOTSPOT - Hotspot 5.0 – Available in: http://lava.cs.virginia.edu/hotspot Access in: 20 Dec. 2013.

HU, J.; OGRAS, U.Y.; MARCULESCU, R., System-Level Buffer Allocation for Application-Specific Networks-on-Chip Router Design, **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, Piscataway: IEEE Press, Vol. 25, Issue 12, p. 2919 - 2933, Dec. 2006.

HYPERTRANSPORT, AMD

Available at:

http://www.amd.com/BR/PRODUCTS/TECHNOLOGIES/HYPERTRANSPORT-TECHNOLOGY/Pages/hypertransport-technology.aspx Access in: 20 Dec. 2013.

IBM CoreConnect Databook - 32-Bit OPB Arbiter Core Databook, 2007.

SA15-5821-01– Available at:

https://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/C05CDD21432D4C1A8725707B0064958D/$file/OPBARB4M.pdf Access in: 20 Dec. 2013.

ITRS: INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS - System Drivers - (2011). Available at:

http://www.itrs.net/Links/2011ITRS/2011Chapters/2011SysDrivers.pdf

INoCs, http://inocs.com/index.php?id=1 Access in: 20 Dec. 2013.

JERGER, E.; PEH, L.; LIPASTI, M., Circuit-Switched Coherence, **Proceedings**… Second ACM/IEEE International Symposium on Networks-on-Chip (NoCS), Washington: IEEE Computer Society, p. 193 - 202, Apr. 2008.

JERGER, N., PEH, L. **On Chip Networks**, Synthesis Lectures on Computer Architecture, Toronto: Morgan & Claypool Publishers, Vol. 4, 141 p., 2009.

JERRAYA, A., WOLF, W. **Multiprocessor System-on-Chips**, 1st ed., [S.l.]: Morgan Kaufmann, 602 p., 2004.

KAHNG, A.B., ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration, **Proceedings**… Design, Automation & Test in Europe - DATE, Leuven: European Design and Automation Association, p. 423 - 428, Apr. 2009.

KIM, D.; ATHIKULWONGSE, K.; LIM, S., A Study of Through-Silicon-Via Impact on the 3D Stacked IC Layout, **Proceedings**… International Conference on Computer-Aided Design - ICCAD, New York: ACM, p. 674 - 680, Nov. 2009.

KOGEL, T., LEUPERS R., MEYR H., **Integrated System-Level Modeling of Network-on-Chip Enabled Multi-Processor Platforms**, 1st ed., Netherlands: Springer, 199 p., 2006.

KUMAR, S. et al., A Network on Chip Architecture and Design Methodology, **Proceedings**… IEEE Computer Society Annual Symposium on VLSI, Washington: IEEE Computer Society, p. 105 - 112, 2002.

KUMAR, A. et al., Toward Ideal On-Chip Communication Using Express Virtual Channels, **IEEE Micro**, Los Alamitos: IEEE Computer Society Press, Vol. 28, no.1, p. 80 - 90, Jan. 2008.

KUMAR, M.P. et al., A Method for Integrating Network-on-Chip Topologies with 3D ICs, **Proceedings**… IEEE Computer Society Annual Symposium on VLSI, Washington: IEEE Computer Society, p. 60 - 65, Jul. 2011.

KUMAR, M. P., MURALI, S, VEEZHINATHAN, K.; Network-on-chips on 3-D ICs: Past, Present, and Future. **Academic Journal**, [S.l.:s.n.], Vol. 29, Issue 4, p. 318 - 335, Jul. – Aug. 2012.

LAFI, W.; LATTARD, D.; JERRAYA, A., An Efficient Hierarchical Router for Large 3D NoCs, **Proceedings**… 21st IEEE International Symposium on Rapid System Prototyping (RSP), [S.l.:s.n.], p. 1 - 5, Jun. 2010.

LAFI, W.; LATTARD, D.; JERRAYA, A., A 3D Reconfigurable Platform for 4G Telecom Applications, Design, **Proceedings**… Automation & Test in Europe Conference & Exhibition (DATE), [S.l.:s.n.], p. 1 - 4, Mar. 2011.

LAI, G.; LIN, X.; LAI, S., GA-Based Floorplan-Aware Topology Synthesis of Application-Specific Network-on-Chip, **Proceedings**… IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), [S.l.:s.n.], Vol. 2, p. 554 - 558, 29-31, Oct. 2010.

LAI, G., LIN, X. Floorplan-Aware Application-Specific Network-on-Chip Topology Synthesis Using Genetic Algorithm Technique. **The Journal of Supercomputing**, [S.l.]:Springer US, Vol. 61, Issue 3, p. 418 - 437, 2011.

LI, F. et al., Design and Management of 3D Chip Multiprocessors Using Network-in-Memory, **Proceedings**… International Symposium on Computer Architecture (ISCA), Washington: IEEE Computer Society, p. 130 - 141, 2006.

LUDIVICI, D. et al., Design Space Exploration of a Mesochronous Link for Cost-Effective and Flexible GALS NoCs, **Proceedings**… Conference on Design, Automation and Test in Europe (DATE), Leuven: European Design and Automation Association, p. 679 - 684, 2010.

LUDOVICI, D., **Technology Aware Network-on-chip Connectivity and Synchronization Design**, 2011, 165 p. Thesis Computer Engineering, Delft University of Technology, Delft, 2011.

MANEVICH, R.; CIDON, I.; KOLODNY, A., Dynamic Traffic Distribution Among Hierarchy Levels in Hierarchical Networks-on-Chip (NoCs), **Proceedings**… IEEE/ACM International Symposium on Networks on Chip (NoCS), [S.l.:s.n.], p. 1 - 8, Apr. 2013.

MANFERDELLI, J.L.; GOVINDARAJU, N.K.; CRALL, C., Challenges and Opportunities in Many-Core Computing, **Proceedings**… IEEE, [S.l.:s.n.], Vol. 96, no.5, p. 808 - 815, May 2008.

MATOS, D. et al., Floorplanning-Aware Design Space Exploration for Application-Specific Hierarchical Network-on-Chip, **Proceedings**… International Workshop on Network on Chip Architectures (NoCArc), New York: ACM, p. 31-36, 2011a.

MATOS, D. et al., A NoC Closed-Loop Performance Monitor and Adapter, **Microprocessors and Microsystems**, Amsterdam: Elsevier Science Publishers, p. 661 – 671, Vol. 37, Issue 6-7, Aug. 2011b.

MATOS, D. et al., Reconfigurable Routers for Low Power and High Performance, **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, Piscataway: IEEE Educational Activities Department, Vol. 19, no.11, p. 2045 - 2057, Nov. 2011c.

MATOS, D. et al., Floorplan-Aware Hierarchical NoC Topology with GALS Interfaces, **Proceedings**… IEEE International Symposium on Circuits and Systems (ISCAS), [S.l.:s.n.], p. 652 - 655, May 2012.

MATOS, D. et al., Hierarchical and Multiple Switching NoC with Floorplan based Adaptability. **Proceedings**… International Conference on Reconfigurable Computing: Architectures, Tools, and Applications (ARC), Berlin: Springer – Verlag, p. 179 – 184, 2013a.

MATOS, D. et al., A Power-Efficient Hierarchical Network-on-Chip Topology for Stacked 3D ICs, **Proceedings**… IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC'13), [S.l.:s.n.], p. 308 – 313, Oct. 2013b.

MEJIA, A. et al., Region-Based Routing: A Mechanism to Support Efficient Routing Algorithms in NoCs, **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, Piscataway: IEEE Educational Activities Department, Vol. 17, no. 3, p. 356 - 369, Marc. 2009.

MENON, R. et al., Switching Activity Minimization in Combinational Logic Design. **Proceedings…** International Conference on Embedded Systems and Applications, ESA/VLSI, [S.l.]: CSREA Press, p. 47 - 53, 2004.

MILLBERG, M., The Nostrum Backbone - a Communication Protocol Stack for Networks on Chip, VLSI Design, 2004. **Proceedings**... International Conference on VLSI Design, Washington: IEEE Computer Society, p. 693 - 696, 2004.

MITIC, M., STOJEEV, M. STAMENKOVIC, Z., Chapter 7: An Overview of SoC Buses. In: OKLOBDZIJA, V. G. **Digital Systems and Applications**: The Computer Engineering Handbook, 2nd Edition, Boca Raton: CRC Press, Chapter 7, 16 p., Nov. 2007.

MO, K. et al., A Hierarchical Hybrid Optical-Electronic Network-on-Chip, **Proceedings**… IEEE Annual Symposium on VLSI, Washington: IEEE Computer Society, p. 327 - 332, Jul. 2010.

MODARRESSI, M.; SARBAZI-AZAD, H.; ARJOMAND, M., A Hybrid Packet-Circuit Switched on-chip Network based on SDM, **Proceedings**… Conference on Design, Automation and Test in Europe (DATE), Leuven: European Design and Automation Association, p. 566 - 569, Apr. 2009.

MODARRESSI, M.; TAVAKKOL, A.; SARBAZI-AZAD, H., Virtual Point-to-Point Connections for NoCs, **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, Piscataway: IEEE Press Vol. 29, no. 6, p. 855 - 868, Jun. 2010.

MODARRESSI, M.; TAVAKKOL, A.; SARBAZI-AZAD, H., Application-Aware Topology Reconfiguration for On-Chip Networks, **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, Piscataway: IEEE Educational Activities Department Vol. 19, no. 11, p. 2010 - 2022, Nov. 2011.

MORRIS, R.; KODI, A.K.; LOURI, A., 3D-NoC: Reconfigurable 3D Photonic on-Chip Interconnect for Multicores, **Proceedings**… International Conference on Computer Design (ICCD), Washington: IEEE Computer Society, p. 413 - 418, 2012.

MUNCH, M. et al., Automating RT-Level Operand Isolation to Minimize Power Consumption in Datapaths, **Proceedings**… Design, Automation and Test in Europe, New York: ACM, p. 624 - 631, 2000.

MURALI, S. et al., Designing Application-Specific Networks on Chips with Floorplan Information. **Proceedings**… IEEE/ACM International Conference on Computer-Aided Design (ICCAD), New York: ACM, p. 355 – 362, 2006.

MURALI, S. et al., Synthesis of Networks on Chips for 3D Systems on Chips, **Proceedings**… Asia and South Pacific Design Automation Conference (ASP-DAC), Piscataway: IEEE Press, p. 242 - 247, Jan. 2009.

NICOPOULOS, C.A. et al., ViChaR: A Dynamic Virtual Channel Regulator for Network-on-Chip Routers, **Proceedings**… International Symposium on Microarchitecture (MICRO). Washington: IEEE Computer Society, p. 333 - 346, Dec. 2006.

NICOPOULOS, C., NARAYANAN V., DAS C. **Network-on-Chip Architectures**: A Holistic Design Exploration. Lecture Notes in Electrical Engineering, 2010 Edition, New York: Springer, Vol. 45, 223 p., 2010.

OETKEN, A. et al., A Bus-Based SoC Architecture for Flexible Module Placement on Reconfigurable FPGAs, **Proceedings…** International Conference on Field Programmable Logic and Applications (FPL), Washington: IEEE Computer Society, p. 234 - 239, 2010.

OKLOBDZIJA, V.G. at al., **Digital System Clocking**: High-Performance and Low-Power Aspects, 1st Edition, Piscataway: IEEE Press, 264 p., Feb. 2003.

OUYANG, J; XIE, Y., LOFT: A High Performance Network-on-Chip Providing Quality-of-Service Support, **Proceedings**… Annual IEEE/ACM International Symposium on Microarchitecture, Washington: IEEE Computer Society, p. 409 - 420, Dec. 2010.

PALERMO, G. et al., Mapping and Topology Customization Approaches for Application-Specific STNoC Designs. **Proceedings**… IEEE International Conf. on Application -Specific Systems, Architectures and Processors (ASAP), [S. l.]: IEEE Computer Society, p. 61 – 68, 2007.

PANADES M., GREINER, A., Bi-Synchronous FIFO for Synchronous Circuit Communication Well Suited for Network-on-Chip in GALS Architectures, **Proceedings**… International Symposium on Networks-on-Chip (NOCS), Washington: IEEE Computer Society, p. 83-94, 2007.

PANDE, P.P.; Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures, **IEEE Transactions on Computers**, Washington: IEEE Computer Society, Vol. 54, no. 8, p. 1025 -1040, Aug. 2005.

PALESI, M.; KUMAR, S.; HOLSMARK, R. A Method for Router Table Compression for Application Specific Routing in Mesh Topology NoC Architectures, **Proceedings**… International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS), Berlin: Springer-Verlag, p. 373 - 384, 2006.

PASSAS, G.; KATEVENIS, M.; PNEVMATIKATOS, D., A 128 x 128 x 24Gb/s Crossbar Interconnecting 128 Tiles in a Single Hop and Occupying 6% of Their Area, **Proceedings**… ACM/IEEE International Symposium on Networks-on-Chip (NOCS), [S.l.:s.n.] p. 87 - 95, May 2010.

PASSAS, G.; KATEVENIS, M.; PNEVMATIKATOS, D., Crossbar NoCs Are Scalable Beyond 100 Nodes, **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)**, [S.l.:s.n.], Vol. 31, no. 4, p. 573 - 585, Apr. 2012.

PASCA V., ANGHEL, L. BENABDENBI, M., Kth-Aggressor Fault (KAF)-Based Thru-Silicon-Via Interconnet Built-in Self-Test Diagnosis, **Journal of Electronic Testing – JET**, [S.l.:s.n.], Vol. 28, Issue 6, p. 817 - 829, 2012.

PASRICHA, S., DUTT, N., **On-Chip Communication Architectures:** System on Chip Interconnect, 1st Edition, San Francisco: Morgan Kaufmann, p. 439 - 466, 2008.

PASRICHA, S., Exploring Serial Vertical Interconnects for 3D ICs. **Proceedings**… In Design Automation Conference (DAC), New York: ACM, p. 581 - 586, 2009.

PASRICHA, S., A Framework for TSV Serialization-aware Synthesis of Application Specific 3D Networks-on-Chip, **Proceedings**… International Conference on VLSI Design (VLSID), Washington: IEEE Computer Society, p. 268 - 273, Jan. 2012.

PAVLIDIS, V. F.; FRIEDMAN, E. G., Chapter 5 – Physical Analysis of NoC Topologies for 3-D Integrated Systems. In: SHEIBANYRAD, A.; PETROT, F., JANTSCH, A. (Eds). **3D Integration for NoC-Based SoC Architectures**. 1st Edition. New York: Springer Publishing Company, 27 p., 2011.

PTM - Predictive technology model, Available at: http://ptm.asu.edu/, Access in: 20 Dec. 2013.

PUTTMANN, C. et al., GigaNoC - A Hierarchical Network-on-Chip for Scalable Chip-Multiprocessors, **Proceedings**… Euromicro Conference on Digital System Design Architectures, Methods and Tools (DSD). Washington: IEEE Computer Society, p. 495 - 502, 2007.

QUICKPATH Interconnect, Intel –

Available at: http://www.intel.com.br/content/www/br/pt/io/quickpath-technology/quick-path-interconnect-introduction-paper.html , Access in: 20 Dec. 2013.

RABAEY, J; CHANDRAKASAN, A.; NIKOLIC, B. Chapter 4: The Wires, **Digital Integrated Circuits**, 2nd Edition, Upper Saddle River: Prentice Hall, 40 p., 2002.

RAHMANI, A.-M. et al., LastZ: An Ultra Optimized 3D Networks-on-Chip Architecture, **Proceedings**… Euromicro Conference on Digital System Design (DSD), Washington: IEEE Computer Society, p. 173 - 180, 2011a.

RAHMANI, A.-M. et al., Congestion Aware, Fault Tolerant, and Thermally Efficient Inter-Layer Communication Scheme for Hybrid NoC-Bus 3D Architectures, **Proceedings**… ACM/IEEE International Symposium on Networks-on-Chip, New York: ACM, p. 65 - 72, 2011b.

RAHMANI, A.-M. et al.; An Efficient Hybridization Scheme for Stacked Mesh 3D NoC Architecture, **Proceedings**… Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), Washington: IEEE Computer Society, p. 507 - 514, 2012.

RENESAS – Available at: http://am.renesas.com/products/soc/assp/mobile/index.jsp Access in 20 Dec. 2013.

ROSTRUP, S., DE STERCK, H., **Parallel Hyperbolic PDE simulation on Clusters**: Cell versus GPU. Computer Physics Communications, [s.n.]:Elsevier, Vol. 181, Issue 12, p. 2164 - 2179, Dec. 2010.

ROYCHOUDHURY, A., MITRA, T., KARRI, S. R., Using formal techniques to Debug the AMBA System-on-Chip Bus Protocol. **Proceedings**… Design, Automation and Test in Europe (DATE), Washington: IEEE Computer Society, p. 828 – 833, 2003.

SAKURAI, T., Approximation of Wiring Delay in MOSFET LSI, **IEEE Journal of Solid-State Circuits**, [S.l.:s.n.], Vol.18, no.4, p. 418 - 426, Aug. 1983.

SEICULESCU, C. et al., SunFloor 3D: A Tool for Networks On Chip Topology Synthesis for 3D Systems on Chips. **Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)**, Piscataway: IEEE Press, Vol. 29, Issue 12, p. 1987 – 2000, 2010.

SEICULESCU, C. et al., Chapter 9: 3D Network on Chip Topology Synthesis: Designing Custom Topologies for Chip Stacks. In: SHEIBANYRAD, A.; PETROT, F. and JANTSCH, A (Eds). **3D Integration for NoC-Based SoC Architectures**, 1st Edition. New York: Springer Publishing Company, p. 193-223, 2011.

SHARMA, M.; KUMAR, D., Wishbone Bus Architecture – A Survey and Comparison. **International Journal of VLSI Design & Communication Systems**. [S.l.:s.n.], Vol. 3, Issue 2, p. 107 - 124, Apr. 2012.

SHEIBANYRAD, A.; GREINER, A. Hybrid-Timing FIFOs to Use on Networks-on-Chip in GALS Architectures, **Proceedings**… International Conference on Embedded Systems and Applications (ESA), [S.l.:s.n.], p. 27-33, 2007.

SHEIBANYRAD, A.; PETROT, F.; JANTSCH, A. Chapter 7 – Asynchronous 3D-NoCs Making Use of Serialized Vertical Links. In: SHEIBANYRAD, A.; PETROT, F., JANTSCH, A. (Eds). **3D Integration for NoC-Based SoC Architectures**. 1st Edition. New York: Springer Publishing Company, 17 p., 2011.

SILVANO, C., LAJOLO, M., PALERMO, G. (Eds.) **Low Power Networks-on-Chip**, 1st Edition, New York: Springer, 300 p., 2011.

SONG, J. et al., An Efficient SoC Test Technique by Reusing On/Off-Chip Bus Bridge, IEEE Transactions on Circuits and Systems, [S.l.:s.n.], Vol. 56, no. 3, p. 554 - 565, Mar. 2009.

SONICS, Available at: http://sonicsinc.com/ Access in: 20 Dec. 2013.

ST, SPEAr Embedded Microprocessors – Available at: http://www.st.com/web/en/catalog/mmc/FM169 Access in: 20 Dec. 2013.

STENSGAARD, M., SPARSO J., ReNoC: A Network-on-Chip Architecture with Reconfigurable Topology. **Proceedings**... International Symposium on Networks-on-Chip, Washington: IEEE Computer Society, p. 55 – 64, 2008.

STRANO, A., LUDOVICI, D., BERTOZZI, D., A Library of GALS Interfaces for Cost-Effective and Flexible MPSoC Design. **Special Issue on Transaction on HiPEAC**, [S.l.:s.n.], Vol. 5, Issue 4, p. 1 - 20, 2011.

SUN, G. et al., A Novel Architecture of the 3D Stacked MRAM L2 Cache for CMPs, **Proceedings**… International Symposium on High Performance Computer Architecture (HPCA), [S.l.:s.n.], p. 239 - 249, 2009.

TAMIR, Y.; FRAZIER, G.L., Dynamically-Allocated Multi-Queue Buffers for VLSI Communication Switches, Washington: IEEE Computer Society, **IEEE Transactions on Computers,** Vol. 41, no. 6, p. 725 - 737, Jun. 1992.

TAN, C., Chapter 1 – Three-Dimensional Integration of Integrated Circuits—an Introduction. In SHEIBANYRAD, A.; PETROT, F. and JANTSCH, A (Eds). In: SHEIBANYRAD, A.; PETROT, F., JANTSCH, A. (Eds). **3D Integration for NoC-Based SoC Architectures**. 1st Edition. New York: Springer Publishing Company, 24 p., 2011.

TEXAS INSTRUMENT - OMAP5, Available at: http://www.ti.com/lsds/ti/omap-applications-processors/omap-5-processors-products.page?paramCriteria=no%3f Access in: 20 Dec. 2013.

THONNART, Y.; BEIGNÉ, E.; VIVET, P. Design and Implementation of a GALS Adapter for ANoC based Architectures. **Proceedings**… IEEE Symposium on Asynchronous Circuits and Systems (ASYNC), Washington: IEEE Computer Society, p. 13 - 22, 2009.

TILERA PRODUCTS - TILE-Gx Processor Family, Available at:

http://www.tilera.com/products/processors/TILE-Gx_Family Access in: 20 Dec. 2013.

TINO, A., KHAN G., Power and Performance Tabu Search Based Multicore Network-on-Chip Design. **Proceedings**… International Conference on Parallel Processing Workshops, [S.l.:s.n.], p. 74 - 81, 2010.

TINO, A., **Multi-Objective Tabu Search Based Topology Synthesis for Designing Power and Performance Efficient NoC Architectures**. 102 p. 2011, Thesis (Masters of Applied Science in the Program of Electrical and Computer Engineering), Ryerson University, Toronto, 2011.

VALDERRAMA, C. et al., FPGA and ASIC Convergence, **Proceedings**… Southern Conference on Programmable Logic (SPL), [S.l.:s.n.], p. 269 - 274, Apr. 2011.

VELENIS, D. et al., Impact of 3D Design Choices on Manufacturing Cost, **Proceedings**… IEEE International Conference on 3D System Integration (3DIC), [S.l.:s.n.], p. 1-5, 2009.

VELLANKI P. et al. Quality-of-Service and Error Control Techniques for Network-on-Chip Architectures, **Proceedings**…Great Lakes Symposium on VLSI. New York: ACM, p. 45-50, 2004.

VISWANATHAN, N.; PARAMASIVAM, K.; SOMASUNDARAM, K., Exploring Hierarchical, Cluster based 3D Topologies for 3D NoC, **Procedia Engineering**, [S.l.:s.n.] Vol. 30, p. 606 - 615, 2012.

WANG, L. et al. A Routing-Table-Based Adaptive and Minimal Routing Scheme on Network-on-Chip Architectures, **Computers & Electrical Engineering**, Tarrytown: Pergamon Press, Vol. 35, Issue 6, p. 846 - 855, 2009.

WEERASEKERA, R. et al., Two-Dimensional and Three-dimensional Integration of Heterogeneous Electronic Systems under Cost, Performance and Technological Constraints, **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)**, Piscataway: IEEE Press, p. 1237 – 1250, 2009.

WELDEZION, A.Y. et al., 3-D Memory Organization and Performance Analysis for Multi-Processor Network-on-Chip Architecture, **Proceedings**… IEEE International Conference on 3D System Integration, (3DIC), [S.l.:s.n.], p. 1 - 7, Sept. 2009.

WIKLUND, D.; LIU, D., SoCBUS: Switched Network on Chip for Hard Real Time Embedded Systems, **Proceedings**… International Symposium on Parallel and Distributed Processing, Washington: IEEE Computer Society, p. 22 – 26, Apr. 2003.

YAN, S.; LIN, B. Application-Specific Network-on-Chip Architecture Synthesis Based on Set Partitions and Steiner Trees. **Proceedings**… Asia and South Pacific Design Automation Conference (ASP-DAC), Los Alamitos: IEEE Computer Society Press, p. 277 – 282, 2008.

YE, T.T.; BENINI, L.; DE MICHELI, G., Analysis of Power Consumption on Switch Fabrics in Network Routers, **Proceedings**… Design Automation Conference (DAC), New York: ACM, p. 524 - 529, 2002.

YING, H.; JAISWAL, A.; HOFMANN, K., Deadlock-Free Routing Algorithms for 3-Dimension Networks-on-Chip with Reduced Vertical Channel Density Topologies, **Journal of Systems Architecture: the EUROMICRO Journal**, New York: Elsevier North-Holland, Vol. 59, Issue 7, p. 528 - 542, 2012.

YING-CHERNG L. et al., A Bidirectional NoC (BiNoC) Architecture With Dynamic Self-Reconfigurable Channel, **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, Piscataway: IEEE Press, Vol. 30, no. 3, p. 427 - 440, Mar. 2011.

YOON, Y. et al., Virtual Channels vs. Multiple Physical Networks: a Comparative Analysis, **Proceedings**… Design Automation Conference (DAC), New York: ACM, p. 162 - 165, 2010.

YU, B. et al., Floorplanning and Topology Generation for Application-Specific Network-on-Chip, **Proceedings**… Asia and South Pacific Design Automation Conference, Piscataway: IEEE Press, p. 535 - 540, Jan. 2010.

XIE, Y.; CONG, J.; SAPATNEKAR, S., **Three-Dimensional Integrated Circuit Design:** EDA, Design and Microarchitectures. 1st Edition, Series: Integrated Circuit and Systems. New York: Springer, p. 189 - 217, 2010.

XU, T. et al., Optimal Number and Placement of Through Silicon Vias in 3D Network-on-Chip, **Proceedings**… International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), [S.l.:s.n.], p. 105 - 110, 2011.

XU, T. et al., A Study of Through Silicon Via Impact to 3D Network-on-Chip Design, **Proceedings**…International Conference On Electronics and Information Engineering (ICEIE), [S.l.:s.n.] p. V1-333 - V1-337, 2010.

ZEFERINO, C. et al. A Study on Communication Issues for Systems-on-Chip, **Proceedings**…Symposium on Integrated Circuits and Systems Design (SBCCI). Washington: IEEE Computer Society, p. 121 – 126, 2002.

ZEFERINO, C.A.; SUSIN, A.A., SoCIN: a parametric and scalable network-on-chip, **Proceedings**… Integrated Circuits and Systems Design (SBCCI), Washington, IEEE Computer Society, p. 169 - 174, 2003.

ZEFERINO, C.A.; KREUTZ, M.E.; SUSIN, A.A., RASoC: a Router Soft-Core for Networks-on-Chip, **Proceedings**… Design, Automation and Test in Europe (DATE), Washington: IEEE Computer Society, Vol. 3, p. 198 - 203, 2004.